Platform for Privacy Preferences (P3P) Specification

W3C Working Draft 26 August 1999

This Version

http://www.w3.org/TR/1999/WD-P3P-19990826

Latest Version:

http://www.w3.org/TR/WD-P3P

Previous Version:

http://www.w3.org/TR/1999/WD-P3P-19990407

Editors:

Massimo Marchiori, W3C/MIT, (massimo@w3.org)

Joseph Reagle, W3C/MIT, (reagle@w3.org)

Abstract

This document describes the Platform for Privacy Preferences (P3P). P3P enables Web sites to express their privacy practices and enables users to exercise preferences over those practices. P3P compliant products will allow users to be informed of site practices (in both machine and human readable formats), to delegate decisions to their computer when appropriate, and to tailor their relationship to specific sites. Site practices that are compatible with a user's preferences can, at the user's option, be accessed "seamlessly". Otherwise users will be notified of a site's practices and have the opportunity to agree to those terms or other terms and continue browsing if they wish.

P3P gives users the ability to make informed decisions regarding their Web experience and the ability to control the use of their information. Sites can use P3P to increase the level of confidence users place in their services, as well as improve the quality of the services offered,

customize content, and simplify site access, offering facilities like auto fill-in of forms, customized profiles, automatic electronic commerce transactions.

Status of This Document

This is the **fifth W3C public working draft** for review by W3C members and other interested parties. This document has been produced as part of the P3P Activity, and will eventually be advanced toward W3C Recommendation status. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress." The underlying concepts of the draft are fairly stable and we encourage the development of experimental implementations and prototypes so as to provide feedback on the specification. However, this Working Group will not allow early implementations to affect their ability to make changes to future versions of this document.

This draft document will be considered by W3C and its members according to W3C process. This document is made public for the purpose of receiving comments that inform the W3C membership and staff on issues likely to affect the implementation, acceptance, and adoption of P3P. A brief annex with the status of some ongoing work is also available. W3C members can access the updated list of pending issues.

Please send comments to www-p3p-public-comments@w3.org (archived at http://lists.w3.org/Archives/Public/www-p3p-public-comments/).

Attention is called to the possibility that implementation of this Technical Report may require use of subject matter covered by patent rights. By publication of this Technical Report, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The W3C shall not be responsible for identifying patent rights for which a license may be required to implement a W3C Technical Report or for conducting inquiries into the existence, legal validity or scope of those patent rights that are brought to its attention.

The P3P 1.0 specification consists of three documents. P3P1.0 compliant implementations must abide by the conformance requirements of each.

Syntax Specification

This is the core and lengthiest specification; it documents the requirements, assumptions, and specifies the P3P protocols, transport methods, and the data structures' syntax and encoding. The actual attribute values for privacy disclosures and data element (names of the information exchanged, like "User.Name.") are specified in the following two documents.

Base Data Set Specification

This document specifies the names of base P3P data elements, sets, and their data types.

Harmonized Vocabulary Specification

This document specifies the English language semantics for privacy related disclosures such as categories, purpose, identifiable use, recipients, and access.

Master Table of Contents

Syntax Specification

Base Data Set and Data Types Specification

Harmonized Privacy Vocabulary Specification

Platform for Privacy Preferences (P3P) Status Notes

26 August 1999

This document was prepared by Lorrie Cranor (lorrie@research.att.com) on behalf of the P3P Specification Working Group. It is an annex to the 26 August 1999 P3P public working draft.

This public working draft is being issued in order to keep the public informed about P3P. However, there are a number of areas where the working group is well aware that further work is needed. These open issues are summarized here. W3C members may also consult the P3P open issues list for more details.

Vocabulary Issues

Questions have been raised about how relationships with credit card companies are reflected in the P3P vocabulary. This and other vocabulary-specific questions need to be examined by the P3P Policy and Outreach Working Group.

Base Data Set Issues

Most of the base data set issues have been resolved in the current draft. Known issues that remain include:

- Need to better document base data set extension mechanism and discuss a few remaining open questions.
- Need to split date type into two sub-types for date and time.

Internationalization Issues

Most of the internationalization issues have been resolved in the current draft. We may add a mechanism that will allow multiple copies of an element to be returned in different languages with the appropriate language attributes.

RDF/XML Syntax Issues

A number of syntax issues have been raised. We need to either fix the RDF syntax or agree to use XML syntax without RDF. Once we solve this problem, we still need to work on a DTD and resolve a number of minor issues. Major syntax changes are still possible at this point.

Protocol Issues

The protocol itself is now fairly well document, but a few open issues still remain:

- Some questions remain about the Limited P3P protocol using the HTML link tag, including whether a propID is needed with a link tag.
- Questions remain about whether there should be restrictions on where a proposal may be stored (i.e. does it have to be stored on the same host or in the same domain as the realm to which it applies?)
- Questions remain about the use of the realm element. It has been proposed that P3P provide syntax for excluding sub-directories from realms. It has also been proposed that the realm element become a header rather than part of the proposal or that it be removed altogether.
- It has been brought to our attention that some servers will not accept HTTP headers bigger than 4k. This may be a concern when the P3P data transfer mechanism is used to return data. We may modify the header syntax to a more compact syntax to address this problem.
- We may also consider removing automatic data transfer (source=agent) from P3P 1.0.

Extension Mechanism

A general P3P extension mechanism has been proposed. This would allow extensions not only to data sets, but to all parts of P3P. This will likely be added to the specification after some additional work. It is likely that this mechanism will include an easy way for implementations that don't wish to recognize extensions to still be compliant.

User Experience Issues

There are a number of issues that impact the P3P user experience. Most of these issues require input from the P3P Policy and Outreach working group before they can be addressed.

Platform for Privacy Preferences (P3P) Syntax Specification

W3C Working Draft 26 August 1999

This Version
    http://www.w3.org/TR/1999/WD-P3P-19990826/syntax
Latest Public Version:
    http://www.w3.org/TR/WD-P3P/syntax
Previous Version:
    http://www.w3.org/TR/1999/WD-P3P-19990407/syntax

Editor:
    Massimo Marchiori, W3C/MIT, (massimo@w3.org)

Abstract

This is the core and lengthiest document of the P3P specification; it documents the requirements and assumptions, and specifies the P3P protocols, transport methods, and the data structures' syntax and encoding.

Status of This Document

This is a subspecification of the P3P specification for review by W3C members and other interested parties. This document has been produced as part of the P3P Activity, and will eventually be advanced toward W3C Recommendation status. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress." The underlying concepts of the draft are fairly stable and we encourage the development of experimental implementations and prototypes so as to provide feedback on the specification. However, this Working Group will not allow early implementations to affect their ability to make changes to future versions of this document.

This draft document will be considered by W3C and its members according to W3C process. This document is made public for the purpose of receiving comments that inform the W3C membership and staff on issues likely to affect the implementation, acceptance, and adoption of P3P.

Send comments to www-p3p-public-comments@w3.org (archived at http://lists.w3.org/Archives/Public/www-p3p-public-comments/).

Table of Contents

1. Introduction

The Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices and enables users to exercise preferences over those practices. P3P compliant products will allow users to be informed of site practices (in both machine and human readable formats), to delegate decisions to their computer when appropriate, and to tailor their relationship to specific sites. Site practices that are compatible with a user's preferences can, at the user's option, be accessed "seamlessly". Otherwise users will be notified of a site's practices and have the opportunity to agree to those terms or other terms and continue browsing if they wish.

P3P gives users the ability to make informed decisions regarding their Web experience and the ability to control the use of their information. Sites can use P3P to increase the level of confidence users place in their services, as well as improve the quality of the services offered, customize content, and simplify site access.

P3P uses [XML] (using the [RDF] data model) for the exchange of structured data and assertions. P3P will support future digital certificate and digital signature capabilities. P3P can be incorporated into browsers, browser plug-ins, servers, or proxy servers that sit between a client and server.

1.1 Problem space

The P3P specification provides mechanisms:

- For a user agent to be informed of a site's data collection and privacy practices.
- For a user agent and service to interact and to come to an agreement satisfactory to both parties; alternatively, for the user agent to notify the user that it could not reach an agreement, and take instruction from the user concerning proposed data exchanges.

1.2 About this specification

This document, along with its normative references and the other chapters of the P3P specification, includes all the specification necessary for the implementation of interoperable P3P applications.

This document includes the main body of the P3P specification. The detailed -- natural language -- semantics of the privacy disclosure vocabulary can be found in [HARMV]. The detailed syntax and datatypes of the base data set is found in [BASEDATA].

Note that while use of the above vocabulary is required for P3P1.0 compliance, the facility of XML-namespaces [XML-name] allows additional or complementary vocabularies to be easily introduced.

We use the following typographical conventions for representing the status of text:

- Schema definition in ABNF notation.
- Brief implementation note.

The ABNF notation used in this specification is specified in RFC2234 and summarized in Appendix 2. However, note that such syntax is only a grammar representative of the XML syntax: all the syntactic flexibilities of XML are also implicitly included; e.g. whitespace rules, quoting using either single quote (') or double quote ("), character escaping, and case sensitivity.

1.3 Operational description and design

When a user encounters a Web service, the Web service may declare its privacy practices and solicit information from the user by sending the user the URI of a *P3P proposal* and an identifier for that proposal called a *propID*. The proposal and propID are transmitted using an extension to the HTTP transport mechanism.

A P3P proposal consists of one or more statements, each of which expresses privacy practices for a set of data elements. The P3P proposal should cover all relevant data elements and practices: if a service wants to collect a data element, it must be mentioned in the proposal. Note that most declarations in P3P are positive, meaning you state what you do, rather than what you do not do.

A core concept to the P3P specification is that of a P3P agreement. A P3P agreement is a proposal agreed to by both the service and user agent. User agents compare the privacy practices specified in a proposal with the user's preferences to determine whether to enter into an agreement. An agreement applies to all data exchanged between the user agent and service within one or more specified realms -- a Web resource or "tree" of Web resources referenced by a single URI.

If a user agent finds a proposal acceptable, it can signal agreement to the web site by returning the proposal identifier. If the proposal includes a request for data to be sent using P3P data transfer methods, the corresponding data is returned with the proposal identifier.

Servers may offer multiple alternative proposals for each realm. Every proposal can have a set of consequences that can be shown to a human user to explain why the suggested practice may be valuable in a particular instance even if the user would not normally allow the practice. User agents should record the agreements reached, indexed by an identifier of the agreement, called the propID. Rather than sending a new proposal URI and propID to the user agent on every contact, a site may send the propID of an existing agreement.

Services may also embed <LINK> tags into HTML content that reference a P3P proposal. This allows services to assert that they follow the practices specified in a P3P proposal without requiring that they use a P3P-compliant server.

There are two notable areas where we substantively provide extensibility mechanisms.

1. How information is solicited from the user. (see `source` attribute)
2. How new data sets are used by services and clients (see `DATA:REF` and namespaces)

Our design is such that applications can be efficiently implemented independent of our assumptions or expectations regarding the latency of multi-round communications, the cacheability of proposals, the use of user agent or server side data repositories, and the size of the agreement repositories.

1.4 Assumptions

P3P makes several assumptions about the environment in which it works and the problem it is trying to solve.

1. P3P allows a service to include identifying information about itself in its proposals. Since the registered owner of a domain name may not correspond to the entity responsible for a service, this information MUST be provided.
2. Strong non-repudiable evidence of identities and agreements will be provided by future versions of the protocol. At this time, no clear/dominant Public Key Infrastructure (PKI) model exists for use in P3P. In the future both users and services may require signatures and certificates.
3. We assume that communication security is achieved through means other than P3P itself [SSL]. Hence, P3P does not provide mechanisms for cryptographically protecting information in storage or transit.
4. P3P agreements are end-to-end: between the user and the service. Intermediaries such as telecommunication providers, internet service providers, proxies and others may be privy to the exchange of data between a service and a user, but their practices are not governed by the agreement between the end parties.
5. While P3P can make profitable use of the HTTP Extension Framework [HTTP-EXT], which is based on [HTTP1.1], P3P should be able to work with [HTTP1.0] servers/proxies.
6. P3P covers all data generated or exchanged via HTTP.

1.5 Terminology

Assuring Party

Within P3P, an assuring party is a legal entity (person or organization) that makes a statement of assurance about a proposal (e.g. that practices are audited, or are in compliance with certain data collection guidelines). Assurance may come from the service itself or from an independent third party. The assuring party must identify what it is attesting to as part of the assurance statement within the proposal, at a specified URI, or as part of the semantic definition of a meta-data schema.

Agreement

A proposal to which both the service and user agent agree. This agreement is applied within the *realm* and is often represented by a propID. The non-repudiability of such agreements will be strengthened by the support of certificate and digital signature capabilities in future versions of P3P; however this is not specified in version 1.0. We do provide the appropriate fields for the inclusion of such tokens within P3P1.0 (e.g. a digital signature from the assuring party.)

Data Element

An individual data entity, such as last name or telephone number. For interoperability, P3P 1.0 specifies a base set of data elements.

Data Category

A significant attribute of a data element or data set that may be used by a trust engine to determine what type of element is under discussion, such as "`Contact Information`." P3P 1.0 specifies 10 base data categories.

Data Set

A known grouping of data element, such as "`user.Home.Postal.`". A set is represented with a trailing period. P3P 1.0 specifies a number of base data sets.

Preference

A rule, or set of rules, that determines what action(s) a user agent will take or allow when involved in a conversation or negotiation with a service. A preference might be expressed as a formally defined computable statement (e.g., the [APPEL] preference exchange language). In this document, preferences govern the types of agreements that can be reached between a user agent and a service.

PropID

A small unit of information used to uniquely identity a P3P proposal (together with the URI the proposal resides). The presence of the propID in the P3P headers is the definitive declaration of which agreements are in effect for a given realm. Whenever a service changes a proposal at the same URI, it MUST change the corresponding propID (for example, a fingerprint of the proposal could be computed, like [MD5]). The propID MUST not be used for any other purpose beyond identifying and referencing P3P proposals (so, for example, ***they cannot be used to maintain user browsing state***).

Proposal

A proposal is a collection of one or more privacy statements together with information asserting the identity, URI, assurances, and disclosures of the service covered by the proposal. A proposal is always created from the point of view of the service and contains identifying information for the service, but it may be created by the user and sent to the server for approval.

Realm

The realm is the experience space from which requests under a given agreement may be issued -- it broadly defines the area to which a proposal applies. It is referenced by one or more URIs. Each URI may name a specific resource or a set of resources qualified by the URI. For instance, in the HTTP URI scheme, a URI ending with an object (home.html) applies to that specific object, a URI that is a path http://www.w3.org/P3P/ references the file system tree below that path. If the proposal is not digitally signed, then each of the URIs must be from a domain that domain-matches the origin server. Domain matching is covered in the HTTP state management mechanism [STATE].

Repository

A mechanism for storing user information under the control of P3P.

Service

A program that issues proposals and (possibly) data requests. By this definition, a service may be a server (site), a local application, a piece of locally active code, such as an ActiveX control or Java applet, or even another user agent.

Statement

A P3P statement is a set of privacy practice disclosures relevant to a collection of data elements, sets, and categories. The enumerated elements may act as an embedded data request. A statement which references no data, does not request any data.

URI

A Uniform Resource Identifier used to identify Web resources. For definitive information on URI syntax and semantics, see [URI].

User Agent

A program whose purpose is to mediate interactions with services on behalf of the user under the user's preferences. A user may have more than one user agent, and agents need not reside on the user's desktop, but *any agent must be controlled by and act on behalf of only the user*. The trust relationship between a user and her agent may be governed by constraints outside of P3P. For instance, an agent may be trusted as a part of the user's operating system or Web client, or as a part of the terms and conditions of an ISP or privacy proxy.

1.6 Conformance requirements

This document specifies requirements over interoperability, feature sets, and policy related semantics. Interoperability requirements constrain implementations such that the protocols operate in accordance with a "shared" protocol state machine and that information is not arbitrarily lost or confounded. This document further specifies requirements over feature sets. This means that while not implementing a feature does *not* break the protocol, it is an abuse of the reciprocating party's -- or user's -- expectation of which features are supported. Furthermore, because of the important policy implications of this application, P3P includes natural language semantics for an XML/RDF schema. To provide an example, an interoperability requirement requires that all parties support the protocol primitives and their responses. Given this requirement, a compliant agent could always return a `SORRY.` However, it abuses the expectations of a service from being able to reach an agreement if all agents are implemented to merely return a sorry. To extend this example, services may not be willing to implement multiple policies or negotiated settlements if no agents exist to do so -- in either case, this lack of a support of a feature does not abuse protocol interoperability.

The breadth of these requirements is atypical but necessary given that P3P is more than a protocol, but an application of existing standards such as HTTP and XML. Furthermore, this document discusses expected features which are not yet implemented and need not be part of the specification itself, but are an important part of the P3P user experience.

The following key words are used throughout the document and should be read as interoperability requirements. This specification uses words as defined in RFC2119 [KEY] for defining the significance of each particular requirement. These words are:

MUST

This word or the adjective "required" means that the item is an absolute requirement of the specification.

SHOULD

This word or the adjective "recommended" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

MAY

This word or the adjective "optional" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

The following table presents a summary of features and operations with respect to their requirements. The context of whether the requirements are placed on the client and/or service is left to the context found in the specification.

Section	**Feature/Operation**	**Key Word**	**Explanation (or Implication if not supported)**
Terminology	propID	MUST	propID uniquely identifies any privacy proposal
Data transport	the HTML LINK tag	MUST	agents and services must be able to locate a proposal in this location
	the HTTP extension mechanism	MUST	agents and services must be able to locate a proposal in this location
XML/RDF encoding	XML parsing	MUST	proposals and data syntax are readily processed or presented to the user. The RDF data model was used to structure the required grammar/syntax but applications need not support RDF if they do not want to take advantage of the RDF data model.
XML/RDF encoding	Full XML and XML-namespace support	MUST	implementations must support full XML and namespaces, not just that used here.
Data References	base data reference syntax	MUST	agents must be able to understand that syntax of solicited information
Harmonized Vocab	harmonized vocabulary	MUST	see the harmonized vocabulary [HARMV] for requirements; they must be followed otherwise the implementation abuses consensus on adequate levels of privacy disclosure
Reason Codes	(OK, SRY-, ERR-, SRY, ERR)	MUST	necessary for reaching agreement.
	propID repository	SHOULD	full proposals can be compactly represented, new negotiation and agreement are not necessary if an agreement already exists.

	data repository	MAY	frequently requested user profile information is stored and managed by the user.
Data Definitions	new schema instantiation	MAY	schemas other than the base set can be instantiated and supported by the agent.
Source Attribute	source attributes "matched-form" and "extension"	MAY	agents support multiple, extensible mechanisms by which information is solicited.

Note that services need not use P3P-compliant servers in order to make P3P proposals available to user agents.

A very simple P3P user agent might do little more than receive P3P proposals and present them to users. Whenever information is requested, such a user agent would prompt the user to make a decision. Thus such a user agent would not make any decisions on behalf of a user or transfer any user data automatically (with the exception of passively generated data such as click-stream data).

User agents that make decisions on behalf of a user or transfer user data automatically must include a user-configurable "trust engine." The trust engine must be able to take a P3P proposal and a set of user preferences -- recorded as a set of rules -- as input and determine what action (seamless accept, seamless reject, informational prompt, warning prompt, or other action) should be taken. User preferences can be encoded in [APPEL] or another language. The preference language should be well-documented or a user interface should be provided for creating rule sets in that language. The user agent must provide a way for users to import rule sets that they create themselves or obtain from others. It should also provide mechanisms for users to create or change rules.

User agents may be initially unconfigured (thus requiring users to configure them before they can be used) or configured with a default rule set. However user agents must not be configured by default to transfer personal information to a service provider without the user's active agreement (seamless accept).

2. Scenarios

The following scenarios are designed to illustrate several possible ways that P3P might be used. Each scenario highlights a different P3P feature. Scenario 1 shows how a basic P3P agreement is established. Scenarios 2 and 3 show what happens when a user returns to a site where an agreement has already been reached. Scenarios 4 highlights the use of the user data repository given existing agreements.

These scenarios illustrate interactions in which no negotiation occurs. However, any scenario could be expanded to one in which a site initially offers multiple alternative proposals.

The table below summarizes the features of each scenario. A "-" in the table indicates a feature about which no assumptions (neutral) are made.

Scenario	Existing Agreement	New Proposal	Data Requested
1	No	Yes	-
2	Yes	No	-
3	Yes	Yes	-
4	-	-	Yes

Scenario 1) No existing agreement, site sends proposal

1. User goes to the home page of CoolCatalog (CoolCatalog/*), she has never been there before but is P3P-compliant.
2. CoolCatalog makes a proposal, including privacy practices, disclosures, and the data elements to which they apply.
3. User agrees to proposal.

Scenario 2) Existing realm agreement

1. User has previously done **scenario 1**.
2. User returns to CoolCatalog and -- following the existing agreement -- is automatically under the current proposal.

Scenario 3) Existing realm agreement, new proposal

1. User has previously completed **scenario 1**.
2. User returns to CoolCatalog and furnishes propID of Agreement(1) on all page requests.
3. User goes to order page on CoolCatalog (CoolCatalog/Order/*) and sends propID of Agreement(1) and the service makes a new proposal because its policies at the order page differ from the rest of the site. In this instance, it wants actual order information.
4. User considers the new proposal, and agrees to it.

Scenario 4) Service wants data from client repository

- Any of **scenarios 1, 2** may apply to the client's first request for content.
- In order to request data, the service has two options:
  1. send the propID of the proposal, with a `SRY-DR` reason code (cf. Section 3.4)
     - if the client *does know* it (by doing lookup against valid agreements entered into to by this client), the client transfers the requested data back to server
     - if the client *does not know* it (either because it never agreed to anything with this service or it simply ran out of space in itslog) the client sends a "sorry" message. Server then sends the complete proposal as follows in option 2. >
     - sends the complete proposal, with a `SRY-DR` reason code (cf. Section 3.4)
       - The client inspects the proposal, matches user preferences or prompts the user for agreement, and either sends data, or breaks off negotiation.

3 Data Transport

3.1 Protocol Model

The protocol model is based on a single round interaction, where the client performs an action (like requesting a proposal), and the server completes the interaction with another action (for example, suggesting a proposal back). Since we follow the rule of thumb that compact data (usually, just a single line) should be placed in an HTTP header, often the response to a request will be a location at which the client must fetch the actual proposal -- just as an HTML client makes several requests to retrieve images in a page. This single interaction (which may involve multiple HTTP requests) can be extended to multiple interactions (or "negotiation") if needed. That is, multiple interactions can be built on top of a single interaction without affecting the simplicity of a single round interaction. It is important to note that the proposed protocol interactions strictly follow the HTTP request/response paradigm in order to properly interact with intermediate caches which may or may not know about P3P.

The basic protocol actions can be summarized as follows.

3.1.1 Client Actions

The client can perform any of the following actions at any point in time when issuing requests:

1. *Request a proposal*
2. *Make a request referencing a known proposal*
3. *Submit data under a known proposal*
4. *Report unacceptable proposals and other errors*

3.1.2 Server Actions

The server can perform the following actions:

1. *Suggest a proposal in response to any request from a P3P-enabled client*
2. *Fulfill a request and accept data under a corresponding proposal*
3. *Reject a proposal (and possibly data) and report errors*
4. *Accept SRY and ERR messages*

3.2 HTTP Extension Framework and P3P

The P3P protocol uses the HTTP Extension Framework [HTTP-EXT]. P3P includes four header extensions, which may be added to any standard HTTP request. The HTTP Extension Framework requires a globally unique URI identifying the extension (the *extension declaration*). The P3P extension declaration is the following URI:

> `http://www.w3.org/TR/WD-P3P/`

The P3P extensions are the following:

`status`

> This describes the status of the P3P message. The status is expressed using a reason code (see Section 3.4).

`propID`

> This contains the propID of the referenced proposal. A propID is meant to be a label that uniquely identifies a proposal at a specified URI (referenced in the `proposal` extension, see below). Whenever a service changes a proposal at the same URI, it MUST change the corresponding propID (for example, a fingerprint of the proposal could be computed, like [MD5]). The propID MUST not be used for any other purpose beyond identifying and referencing P3P proposals (so, ***they MUST not be used to maintain user browsing state***).
> When returning to a web site, user agents SHOULD verify that the propID has not changed. If the propID changes, the user agent SHOULD request the proposal again and re-evaluate it. As an added safeguard, user agents MAY periodically request the proposal again to make sure the proposal was not changed without changing the propID.

`proposal`

> This contains the URI where the proposal can be fetched.

`data`

> This contains, line by line, the XML message encoding the data that the client sends to the server under a specific agreement.

3.3 Protocol Actions

This section describes the actions P3P-compliant clients and servers take to carry out the P3P protocol as an extension to the HTTP protocol. This section includes references to the various error codes that must be reported in carrying out the protocol. These codes are defined and discussed in more detail in section 3.4.

3.3.1 P3P Client Actions

P3P clients MUST be able to:

Request a proposal

> To request a proposal from a P3P-compliant server, a P3P-enabled client MUST issue a request that includes either an optional or mandatory P3P HTTP header extension, for example:
>
> > `Opt: "http://www.w3.org/TR/WD-P3P/"`
> > or `Man: "http://www.w3.org/TR/WD-P3P/"`
>
> Generally the `Opt` header should be used. However, if the client wishes to express that it requires P3P support, the mandatory extension may be used along with the methods `M-GET`, `M-POST`, etc. A P3P-compliant server should not treat `Man` headers any differently than it treats `Opt` headers.
>
> Any request that contains a P3P HTTP header extension is an implied request for a proposal. A P3P-compliant server will always send the `propID` and `proposal` P3P extension headers when a relevant proposal is available.
>
> If a client wishes to *just* request a proposal, without content, the client may issue an `OPTIONS` or `HEAD` request.
>
> If a client wishes to simultaneously request a proposal and content, the client SHOULD issue a `GET` request (or other relevant request). If the server is willing to provide content regardless of whether a P3P agreement is reached, the server will provide the appropriate content along with the `propID` and `proposal` P3P extension headers. If the server is not willing to provide content until an agreement is reached, the server will respond with the `SRY-AR` code in a P3P `status` header and possibly an `HTTP 510` error code. If the proposal is acceptable the client SHOULD repeat the original request, this time including an extension header with the relevant propID.
>
> Note, when a user directs a P3P-compliant user agent to reach agreements with web sites before exchanging data, that client should not transfer data to a URI for which there is no corresponding P3P agreement. Thus, while a `POST` request is also an implicit request for a proposal, it should not be used that way by clients configured not to transfer data without agreement. Instead an `OPTIONS` or `HEAD` request should be made. Once the propID has been obtained, then the `POST` request can be made.

Likewise, clients configured not to transfer data without an agreement should issue an OPTIONS or HEAD request before sending a GET request that includes form data (generally such a request would include a question mark) to a realm with which no agreement exists. In addition, there may be other cases where client implementations may need to be careful not to transfer data prior to reaching an agreement.

Make a request referencing a known proposal

If the client already knows the propID of an appropriate proposal that it finds acceptable, it can make a request that includes a propID header. The client may include a propID header in any standard HTTP request. If the propID is valid, a P3P-compliant server should try to fulfill the request. If the propID is invalid, a P3P-compliant server is expected to respond with an appropriate error in a P3P status header and possibly an HTTP 510 error code.

Submit data under a known proposal

If the client is making a request for which the corresponding proposal requests that data be submitted using P3P methods, the client MUST be able to submit data in a P3P data header attached to that request. All P3P-compliant clients MUST be able to submit data in conjunction with GET and POST requests. They MAY also be able to submit data in conjunction with other HTTP requests.

The client SHOULD submit any required data the first time it includes a given propID in a request. If a client submits a request including a propID and does not submit the required data, the server may request the data by sending a P3P SRY-DR in a status header and possibly also an HTTP 510 error. If the client receives a SRY-DR it SHOULD repeat the request with the required data if it still wishes to access the requested resource.

Report unacceptable proposals and other errors

If none of the proposals offered by a service in proposal headers are acceptable (because the policy is unacceptable, the client doesn't have the requested data, or because of a technical problem such as an invalid propID or syntax error), the client SHOULD respond by repeating the request with the appropriate SRY or ERR code in an extension header. However, clients SHOULD NOT repeat successful POST requests or other requests with side effects. In addition, if the client no longer wishes to access the requested resource, it MAY choose not to respond to an unacceptable proposal. If the service offers multiple proposals and the client finds at least one acceptable, it need not report a SRY or ERR for the others. If it finds all proposals unacceptable and wishes to respond, it SHOULD report a SRY or ERR code for each. Clients MUST take precautions to avoid getting into infinite loops in which they report errors, receive a response from the service, and report errors again.

3.3.2 P3P Server Actions

P3P servers MUST be able to:

Suggest a proposal in response to any request from a P3P-enabled client

P3P-compliant servers MUST include propID and proposal headers in all responses to requests from P3P-enabled clients when a relevant proposal is available. Servers MAY indicate the availability of multiple proposals for a particular resource by including multiple propID and proposal headers in their responses.

When a client requests proposals or human readable privacy policies by issuing a GET request to a URI indicated by a proposal header, P3P LINK tag, or a discuri tag within a proposal, the server MUST fulfill the request without requiring an agreement.

Fulfill a request and accept data under a corresponding proposal

When a request is made that includes a propID header, servers MUST verify that the propID is valid. If the propID is valid, they SHOULD try to fulfill the request. P3P-compliant servers MUST have the ability to accept data submitted in data headers as part of GET and POST requests. They MAY also accept data as part of any HTTP requests.

Reject a proposal (and possibly data) and report errors

When a request is made with an invalid propID, the server MUST respond with the appropriate error code in a status header. If data is submitted with an invalid propID, the data MUST be thrown away. If the request is not fulfilled, the server SHOULD respond with an HTTP 510 error code in addition to the P3P status header.

When reporting errors involving multiple proposals, the server may either report a single error and list all of the propIDs to which that error applies, or it may report an error for each propID. If different errors apply to each propID, they MUST be reported seperately.

Accept SRY and ERR messages

P3P-compliant servers MUST be able to accept SRY and ERR messages from the client and log or report these messages to the server operator.

3.3.3 Protocol Example

The following is an example of how interactions between P3P-compliant clients and servers will typically proceed. Examples of client and server actions are shown in the shaded boxes.

1. Client requests resource using `GET` or `POST` or whatever method it pleases. Client also sends P3P extension header to indicate that it is P3P-enabled.

```
GET coolcatalog.com/index.html HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"
```

2a. If server does not a require P3P agreement, the server returns content and `propID` and `proposal` headers. Multiple sets of `propID` and `proposal` headers may be used to indicate that the server offers a choice of multiple alternative proposals. (Client may optionally fetch and evaluate each proposal before displaying the content.) [If all proposals are unacceptable skip to step 3b, otherwise skip to step 5]

```
HTTP/1.1 200 OK
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-proposal: http://coolcatalog.com/P3PProposal1.xml
Opt: "http://www.w3.org/TR/WD0P3P/"; ns=2
2-propID: 9Pfx3KzqthhRfWxxW1gLnQ==
2-proposal: http://coolcatalog.com/P3PProposal2.xml
. . .
Content-Type: text/html
. . .
```

2b. If the server requires a P3P agreement, the server returns one or more `propID`/`proposal` headers, a `status: SRY-AR` header, and optionally an `HTTP 510` error code.

```
HTTP/1.1 510 Not Extended
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-proposal: http://coolcatalog.com/P3PProposal1.xml
1-status: SRY-AR
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=2
2-propID: 9Pfx3KzqthhRfWxxW1gLnQ==
2-proposal: http://coolcatalog.com/P3PProposal2.xml
2-status: SRY-AR
. . .
```

(If the client needs to evaluate the proposal, it fetches it by issuing a `GET` request to the URI indicated by the `proposal` header. After fetching the proposal, the client may optionally interact with the user to determine whether the proposal is acceptable and/or display information to the user.)

```
GET coolcatalog.com/P3PProposal1.xml HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"
```

3a. If the proposal is acceptable, the client resubmits the request, this time including propID and any requested data. (If more than one proposal is available, the client should select only one and include the corresponding propID in its response.)

```
GET coolcatalog.com/index.html HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-data: <TXD xmlns="http://www.w3.org/TR/WD-P3P/syntax"
  xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
  xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
1-data: <DATA:REF name="user.name.first" value="Sheila">
1-data: <DATA:REF name="user.name.second" value="Doherty">
1-data: </TXD>
. . .
```

3b. If the proposal is not acceptable or if there is some other error, the client repeats the request with the appropriate `SRY` or `ERR` header (unless the request was a successful `POST` request or some other request that would have an undesirable side effect if repeated). [return to step 2 but client should give up after a small number of failed attempts to avoid infinite loop]

```
GET coolcatalog.com/index.html HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-status: SRY-PR
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=2
2-propID: 9Pfx3KzqthhRfWxxW1gLnQ==
2-status: SRY-PR
```

Alternatively, this may be written:

```
GET coolcatalog.com/index.html HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-propID: 9Pfx3KzqthhRfWxxW1gLnQ==
1-status: SRY-PR
```

4a. If the propID is valid and there are no problems, the server returns content plus `proposal`/`propID` headers.

```
HTTP/1.1 200 OK
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-proposal: http://coolcatalog.com/P3PProposal1.xml
. . .
Content-Type: text/html
. . .
```

4b. If there is some other problem (for example, the client did not actually send the required data), the server returns a `status` header with the appropriate `SRY` or `ERR` code, and `proposal`/`propID` headers. The server may optionally return an `HTTP 510` error code if the request is not fulfilled. [return to step 3 to try again, but client should give up after a small number of failed attempts to avoid infinite loop]

```
HTTP/1.1 510 Not Extended
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Xorq/ZJbTpVaFRD4qfA==
1-status: SRY-DR
2-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
2-proposal: http://coolcatalog.com/P3PProposal1.xml
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=2
3-propID: 9Pfx3KzqthhRfWxxW1gLnQ==
3-proposal: http://coolcatalog.com/P3PProposal2.xml
. . .
```

[time passes]

5. The client requests the same resource as before. It includes a `propID` header in order to indicate that it agrees to the proposal.

```
GET coolcatalog.com/index.html HTTP/1.1
Host: coolcatalog.com
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: "Ed3Oorq/ZJbTpVaFRD4qfA=="
. . .
```

6a. If the agreement is still valid and the server doesn't need the data again (or if the server does not require a P3P agreement), it simply returns the content, together with `proposal`/`propID` headers (as in step 4a).

```
HTTP/1.1 200 OK
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-proposal: http://coolcatalog.com/P3PProposal1.xml
. . .
Content-Type: text/html
. . .
```

6b. If the agreement is still valid, but the server needs the data again, it asks the client to resubmit data by sending a `status: SRY-DR` header and optionally an `HTTP 510` error code. [continue to step 3]

```
HTTP/1.1 510
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-proposal: http://coolcatalog.com/P3PProposal1.xml
1-status: SRY-DR
. . .
```

6c. If there is some other problem, the server returns a `status` header with the appropriate `SRY` or `ERR` codes, `proposal`/`propID` headers, and optionally an `HTTP 510` error code. [return to step 5 to try again, but client should give up after a small number of failed attempts to avoid infinite loop]

```
HTTP/1.1 510
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=1
1-propID: Ed3Oorq/ZJbTpVaFRD4qfA==
1-status: ERR-IF
1-proposal: http://coolcatalog.com/P3PProposal1.xml
. . .
```

3.4 Error messages

Each protocol action can trigger an error, due to various reasons. For example, the proposal could be malformed, or the referenced agreement could be unknown, and so forth. An *error message* is issued using the following P3P extensions:

- `status` to the appropriate reason code(s) (if there are no errors to report, then this extension is omitted).
- `propID` to the corresponding propID that is referenced (if there is no such propID, then this extension is omitted).
- ***(optionally)*** `proposal` to the URI where the referenced proposal can be fetched

Reason codes are sent in the status extention headers and indicate the state of a proposal or data -- potentially -- in the context of a previous interaction. P3P extensions must only include status headers when they are reporting errors. In all the other cases (that is to say, when receipt of the contents of the proposal or data extension have been completed successfully), no status header must be included. Applications that do not understand the status extension must ignore the p3p-message.

Reason Token	*Description*	*Sent by*	*Explaination*
SRY-AR	the **a**greement is **r**equired	server	Sent (together with a propID) if a prior agreement (corresponding to the propID) is required. Clients should evaluate the proposal (this might require them to download it first) and according to the user's preferences resubmit (with or without data), send a SRY(-PR) or end the communication.
SRY-AU	the **a**greement is **u**nkown	server	Sent if a client sent an unkown propID along with its request. If a new proposal is available that requires agreement, the site would also have to send a SRY-AR (see above). If no "SRY-AR" is present, client can re-submit the request without referencing any propID.
SRY-DR	the **d**ata is **r**equired	server	Sent if the client agreed to an agreement (referenced by the corresponding propID), but did not send the proper data. Clients would need to resubmit the same request but also include the required (according to the proposal) data, if they want to access the resource.
SRY-DU	the **d**ata is **u**navailable	client	Sent if a client cannot provide the requested data, because they are not available.
SRY-PR	the **p**roposal is **r**ejected	client	Sent if the proposal is unacceptable. Servers can log this and optionally redirect the client or offer a different page/proposal.
SRY	**s**or**ry**	server or client	Sent if receipt of the contents of the proposal or data extension failed, despite syntactically correct, and all the other SRY-codes are not applicable, or for privacy reasons (see afterwards).
ERR-IF	**i**nvalid **f**ormat	server or client	The P3P message that was received was syntactically invalid or semantic garbage.
ERR-TF	data **t**ransfer **f**ailed	server or client	the transfer request was received, and it was syntactically valid, and covered by a valid agreement, but the data could not be stored for some reason
ERR	**err**or	server or client	a request was not understood or received properly. This code can be sent when all the other ERR- codes are not applicable, or for privacy reasons (see afterwards).

Note that if, for privacy reasons, the client does not want to divulge its reasons for refusing a proposal, the client can reply with the generic SRY code, rather than with a more specific SRY- code. The same holds for the error messages (ERR can be used rather than a specific ERR-), although it is advisable that ERR-IF and ERR-TF are used whenever possible.

Server MUST at least support sending out SRY-AR and ERR. Clients MUST understand at least the three specific SRY- error codes that can be sent by the server, that is to say they MUST be able to (if the user wants) accept a proposal (by sending the propID) and send data if necessary (again at the user's discretion). In case of SRY-AU, clients MUST be able to resubmit the request without referencing a (probably outdated or unknown) propID.

Finally, note than more than one reason codes can be used to specify an error status; for example, a client as a response to proposal suggestion by the server could return `SRY-DU` to indicate some requested data is not available, and `ERR-IF` to indicate that some part of the proposal was syntactically incorrect too:

```
HTTP/1.1 200 OK
Opt: "http://www.w3.org/TR/WD-P3P/"; ns=14
14-status: SRY-DU
14-status: ERR-IF
14-propID: 94df1293c1p8
14-proposal: http://www.privacy.org/newP3PProposal
...
[Content]
```

3.5 Limited Protocol

A limited version of the P3P protocol may be executed without using the HTTP header extension mechanism. Instead, servers may serve HTML content with embedded `LINK` tags that indicate the location of relevant P3P proposals. This limited protocol requires a P3P-aware client, but does not require a P3P-aware server (content may be modified to include the embedded link tags without requiring any changes to the way the server operates).

A service that uses the limited protocol may declare that it will abide by a given P3P proposal by embedding a `LINK` tag of the following form in the `<HEAD>` area of an HTML document:

```
<LINK rel="p3p:propID" href="URI">
```

URI indicates the location of the P3P proposal and "p3p:*propID*" is a relation name used to distinguish this special P3P link and to encode the propID of the proposal. For example, if a proposal at http://www.privacy.org/P3PProposal has propID `x3tYwafhfKSqGV0Q+eSOZw==`, we would use `<LINK rel="p3p:x3tYwafhfKSqGV0Q+eSOZw==" href="http://www.privacy.org/P3PProposal">`.

P3P-compliant clients MUST search for such `LINK` tags whenever they make a request that results in the return of HTML content without the inclusion of `propID` and `proposal` headers. In addition, P3P-compliant clients MAY search for `LINK` tags embedded in HTML content at any time. If a client finds a P3P `LINK` tag in addition to `propID` and `proposal` headers in the same document, it should consider them to reference multiple alternative proposals.

The limited protocol is not suitable for services that wish to use P3P methods for data collection, offer a choice of proposals, or require explicit agreement to proposals.

The following is an example of how interactions between P3P-compliant clients and non-P3P-compliant servers will typically proceed. Examples of client and server actions are shown in the shaded boxes.

1. Client requests resource using `GET` or `POST` or whatever method it pleases. Client also sends P3P extension header to indicate that it is P3P-enabled.

```
GET coolstore.com/index.html HTTP/1.1
Host: coolstore.com
Opt: "http://www.w3.org/TR/WD-P3P/"
```

2. Server is not P3P enabled so it ignores the P3P extension header (unless it is a mandatory extension and/or the server is not extension enabled, in which case the server reports back that it is not P3P enabled) and sends the requested content.

3. Client checks the returned content for an embedded link tag with attribute rel="p3p:*propID*". If no such link tag exists and the requested resource is not in a realm with which the client already has an agreement, the client can assume there is no P3P proposal available and behaves accordingly. If the link tag does exist, the client may fetch the proposal by issuing a `GET` request to the URI indicated by the `href` attribute of the link tag, evaluate the proposal, and act according.

Note, in all cases the appropriate behavior after evaluating a proposal and/or determining that no proposal exists depends on the user's preferences and how the client is configured. Clients may proceed with a request, display information, prompt the user, or take some other action.

4. P3P markup and processing

This section describes the syntax and semantics of P3P proposals as well as for blocks of user data transmitted between a user agent and a service. *A more complete English language description of the terms used in a proposal may be found in [HARMV].*

We begin with an example of an English language privacy policy and a corresponding P3P proposal in Section 4.1. P3P proposals include general assertions that apply to the entire proposal as well as specific assertions -- called *statements* -- that apply only to the handling of particular types of data referred to by *data references*. Section 4.2 describes the `proposal` element and proposal-level assertions. Section 4.3 describes the `TXD` element that is used to transfer data blocks. Section 4.4 describes statements and data references.

In the sections that follow we introduce a number of XML elements. Each element is given in <> brackets, followed by a list of valid attributes and an XML name space (xmlns). All listed attributes are optional, except when tagged as *mandatory*.

4.1 Example proposal

4.1.1 English language proposal

The following is an example of an English-language privacy policy that we would like to encode as a P3P proposal.

> CoolCatalog makes the following statement for the Web pages at http://www.CoolCatalog.com/catalog/.
>
> We use cookies and collect, via a web form, your gender and (optionally) your home address to customize our entry catalog pages and for our own research and product development. We do not use this information in identifiable form.
>
> We also maintain server logs that include information about the what pages on our site are visited and the types of browsers our visitors use. We use this information in order to maintain and improve our web site. We do not use this information in an identifiable way.
>
> We do not provide access capabilities to information we may have from you, but we do have retention and opt-out policies, which you can read more about at our privacy page http://www.CoolCatalog.com/PrivacyPractice.html. The third-party PrivacySeal.org provides assurance that we abide by this agreement.

The following is a more formal description, using the P3P element and attribute names, with their numeric values in parentheses:

> Entity: http://www.CoolCatalog.com/
> Realm: http://www.CoolCatalog.com/catalog/
> Disclosure URI: http://www.CoolCatalog.com/PrivacyPractice.html
> Access to Identifiable Information: none
> Assurance: PrivacySeal.org
> Other disclosures: Change agreement, retention
>
> We collect:
>   dynamic.cookies
>   user.gender
>   user.home. (optional)
> For purpose: Customization of the site to individuals, research and development
> Identifiable use: No
> Recipients: Only ourselves and our agents
> Consequence: A site with clothes you would appreciate
>
> We collect:
>   dynamic.clickstream.server
>   dynamic.http.useragent
> For purpose: Web site and system administration, research and development
> Identifiable use: No
> Recipients: Only ourselves and our agents

4.1.2 XML/RDF encoding of proposal

The following piece of [RDF] captures the information as expressed above. P3P proposals are statements that are properly expressed according to the syntax model of [RDF] as well as well-formed XML. However, there are two assumptions, used to make the proposal shorter, that slightly differentiate a P3P proposal from standard RDF. If the XML/RDF data is homogeneously P3P then the enclosing RDF tags may be optionally omitted. The proposal syntax will be explained in more detail in the sections that follow.

```
<RDF:RDF xmlns:RDF="http://www.w3.org/TR/WD-rdf-syntax">
<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax"
 xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
 xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata"
 entity="http://www.CoolCatalog.com">
  <ASSURANCE org="http://www.PrivacySeal.org"
   text="third party" image="http://www.PrivacySeal.org/Logo.gif"/>
  <REALM uri="http://www.CoolCatalog.com/catalog/">
  <VOC:DISCLOSURE discuri="http://www.CoolCatalog.com/PrivacyPractice.html"
   access="none" retention="yes" change_agreement="yes"/>
  <USES>
  <STATEMENT VOC:id="nonid">
     <CONSQ v="a site with clothes you would appreciate"/>
    <VOC:RECPNT v="ours"/>
     <VOC:PURPOSE v="uniqueid"/>
     <VOC:PURPOSE v="financial"/>
     <DATA:REF name="dynamic.cookies" source="service"/>
     <DATA:REF name="user.gender" source="matched-form"/>
     <DATA:REF name="user.home." optional="yes" source="matched-form"/>
  </STATEMENT>
  </USES>
  <USES>
  <STATEMENT VOC:id="nonid">
    <VOC:RECPNT v="ours"/>
    <VOC:PURPOSE v="admin"/>
    <VOC:PURPOSE v="develop"/>
    <DATA:REF name="dynamic.clickstream.server" source="service"/>
    <DATA:REF name="dynamic.http.useragent" source="service"/>
  </STATEMENT>
  </USES>
</PROP>
</RDF:RDF>
```

4.2 Proposals

This section defines the key elements, attributes and processing heuristics for operating on a proposal. All proposals are encoded using [UTF-8].

4.2.1 The PROP element

`<PROP>` (`xmlns="http://www.w3.org/TR/WD-P3P/syntax"`)

includes one or more statements. Each statement includes a set of disclosures as applied to a set of data elements.

`entity` (***mandatory attribute***)

URI referencing a domain name and path that can be associated with the legal entity making the representation of the privacy practices contained in the proposal.

```
[1]PROP-msg=(`<RDF:RDF xmlns:RDF="http://www.w3.org/TR/REC-rdf-syntax/">`
            proposal
            "</RDF:RDF>")|
           proposal
```

```
[2]proposal=`<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax"
            xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
            xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata"`
           " entity=" quoted-URI
           ">"
           1*statement-block
           disclosure
           [assurance]
           1*realm
           `</PROP>`
```

4.2.2 The `REALM` element

`<REALM>` (`xmlns="http://www.w3.org/TR/WD-P3P/syntax"`)

describes the *realm* of the proposal (the URIs to which the proposal applies)

`uri` (***mandatory attribute***)

the URI to which the proposal applies

```
[3]   realm       =  "<REALM"
                      " uri=" quoted-URI
                     "/>"
```

The URIs specified by REALM define the scope of the agreement (the "realm"). This information is used by the user agent, for example, to determine whether there is an existing agreement with the service. Each of the Realm URIs should "domain-match" [STATE] the domain of the server.

Each URI may name a specific resource or a set of resources qualified by the URI. In the HTTP URI scheme, a URI ending in "/" references the file system tree below that path (for example, http://www.w3.org/P3P/" would apply to "http://www.w3.org/P3P/foo/schema.html" as well); on the other hand, a URI that does not end in "/" would apply only to that specific resource (for example, "http://www.w3.org/P3P/data.html" wot apply to "http://www.w3.org/P3P/foo/schema.html").

User agents should record the realms with which an agreement has been reached and the corresponding propIDs and store them for as long as the agreements remain valid (or as long as feasible). When returning to any part of a realm previously visited, the user agent should include the relevant propID in its requests to the service.

Schemes

P3P is designed around HTTP and related schemes (such as HTTPS). Specifically in the case of HTTP and HTTPS, agreements reached using HTTP implicitly cover URI requests that use the HTTPS scheme (but not vice versa). For example, if a user reaches an agreement with realm "http://www.romulus.com/", then "https://www.romulus.com/" is considered part of the realm. However, if the user reached an agreement with realm "https://www.romulus.com/", then "http://www.romulus.com/" is not subject to the agreement. Agreements may be reached regarding realms that include other schemes such as FTP; however, the agreement must be reached by conducting the protocol over HTTP (or HTTPS) or through links embedded in HTML content. The protocol may be extended in the future to cover data exchange and negotiation using other schemes, such as FTP or NNTP.

4.2.3 The `DISCLOSURE` element

`<DISCLOSURE>` (`xmlns="http://www.w3.org/TR/WD-P3P/syntax"`)

simple disclosures regarding service access capabilities, and binary values as to whether the entity makes disclosures regarding changing the agreement (opt-out) for data already collected and how long data is retained.
`xmlns="http://www.w3.org/TR/WD-P3P/vocab"`

`discuri` (***mandatory attribute***)

URI of the natural language privacy statement of the proposal, which should include information on how to contact the service with questions or concerns.

`access`

the ability of the individual to view identifiable information and address questions or concerns to the service provider.

`retention`

does the site make its policy regarding retention known at its discuri?

`change_agreement`

does the site make its policy regarding change agreement known at its discuri?

The values of [5] are provided for reference only. The normative specification is in the harmonized vocabulary [HARMV] document.

```
[4]disclosure       ="<VOC:DISCLOSURE"
                     " discuri=" quoted-URI
                     " access=" `"` access-disclosure `"`
                     [" retention=" `"` yesno `"`]
                     [" change_agreement=" `"` yesno `"`]
                     "/>"
[5]access-disclosure="nonident" | ; Identifiable Data is Not Used
                     "contact"  | ; Identifiable Contact Information
                     "other"    | ; Other Identifiable Information
                     "contact_and_other" | ; Identifiable and other contact information
                     "none"       ; None
[6]yesno            ="yes" | "no"
```

4.2.4 The ASSURANCE element

<ASSURANCE> (xmlns="http://www.w3.org/TR/WD-P3P/syntax")

describes a services that attests that the entity will abide by its proposal, follows guidelines in the processing of data, or other relevant assertions.

service

URI of the assurance service

text

short textual description of the type of assurance service (e.g., third party, legal, etc.)

image

URI of an image logo of the assurance service

width

width in pixels of the image logo

height

height in pixels of the image logo

alt

very short textual description of the image logo

```
[7]  assurance     = "<ASSURANCE"
                      " service=" quoted-URI
                      [" text=" quoted-string]
                      [" image=" quoted-URI
                      [" width=" `"` number `"`]
                      [" height=" `"` number `"`]
                      [" alt=" quoted-string]
                     "/>"
```

Note that there can be multiple assurance services, specified via multiple occurrences of ASSURANCE.

4.2.5 Semantics of a Proposal

If a website links to a P3P proposal, the website is stating that it is willing to honor that proposal for any user agent that sends a reference to that proposal to the website in the header of its http requests.

If a website links to multiple P3P proposals, this means that the website is only bound by the P3P proposal that the user agent chooses out of the multiple proposals. The user agent signals that it has chosen a specific P3P proposal by sending a reference to that proposal in the headers.

A website is not bound by any P3P proposal if the user agent does not reference a P3P proposal in the headers of its requests, even if that website has only one P3P proposal. Therefore user agents that do not support p3p do not bind the website to any particular P3P proposal. Furthermore, the first request a user agent makes of a website is unlikely to be covered by any P3P proposals, since the user agent probably does not yet know what P3P proposal to reference in the headers of the request.

Websites with web servers that do not support p3p should avoid linking to multiple proposals, as the website probably has no way of determining which proposal the user agent has chosen. If, despite this advice, a website links to multiple proposals, the website is bound to somehow honor whichever proposal the user agent has selected, no matter what difficulty this places on the website.

4.3 Data Transmission

The client sends data referencing a specific agreement to the service (see Section 3.3.2) by transmitting an XML/RDF message in the header.

`<TXD>` (`xmlns="http://www.w3.org/TR/WD-P3P/syntax"`)

indicates that a block of data is returned under an agreement referenced by the propID

```
[8]TXD-msg  =(`<RDF:RDF>` txd-block `</RDF:RDF>`) |
             txd-block

[9]txd-block=`<TXD xmlns="http://www.w3.org/TR/WD-P3P/syntax"
                  xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
                  xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">`
            *(data-reference)
            `</TXD>`
```

When TXD messages are sent as HTTP header extensions, line breaks should be inserted after each closing angle bracket `>`. In addition, the appropriate prefix must be inserted on each line. For example:

```
42-data: <TXD xmlns="http://www.w3.org/TR/WD-P3P/syntax"
xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
42-data: <DATA:REF name="user.name.first" value="Sheila"/>
42-data: <DATA:REF name="user.name.Last" value="Doherty"/>
42-data: <DATA:REF name="user.name.Title" value="Miss"/>
42-data: </TXD>
```

4.4 Statements

Statements describe data practices that are applied to particular types of data.

4.4.1 The `STATEMENT` element

`<STATEMENT>` (`xmlns="http://www.w3.org/TR/WD-P3P/syntax"`)

data practices as applied to data elements.

`id` **(*mandatory attribute*)** (`xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"`)

specifies if data used in a way that is personally identifiable (including linking it with identifiable information about you from other sources)

`consq`

consequences that can be shown to a human user to explain why the suggested practice may be valuable in a particular instance even if the user would not normally allow the practice.

```
[10]statement-block= "<USES>"
                    "<STATEMENT"
                    " VOC:id=" `"` idvalue `"`
                    ">"
                    *(consequence)
                    *(purpose)
                    *(recipient)
                    *(data-reference)
                    "</STATEMENT>"
                    "</USES>"
```

The values of [11-13] are provided for reference only. The normative specification is in the harmonized vocabulary [HARMV] document.

```
[11]purposevalue   = "current" | ; Completion and Support of Current Activity
                     "admin"   | ; Web Site and System Administration
                     "custom"  | ; Customization of the Site to Individuals
                     "develop" | ; Research and Development
                     "contact" | ; Contacting Visitors for Marketing of Services or Products
                     "other" [" (" string ")"] ; Other Uses

[12]recipientvalue = "ours" | ; only ourselves and our agents
                     "same" | ; organizations following our practices
                     "other" | ; organizations following different practices
                     "published"   ; unrelated third parties or public forum

[13]idvalue        = "nonid" | ; non identifiable
                     "id"      ; identifiable
```

4.4.2 The `CONSQ` element

`<CONSQ>` (`xmlns="http://www.w3.org/TR/WD-P3P/vocab`")

consequences that can be shown to a human user to explain why the suggested practice may be valuable in a particular instance even if the user would not normally allow the practice.

`v` **(*mandatory attribute*)**

the corresponding value

xml:lang

the language in which the consequence is expressed

Multiple consequences can be indicated using multiple `CONSQ` elements, each with one of the consequences. This way, multi-lingual versions of the consequences can be encoded, using the `xml:lang` attribute.

```
[14]consequence= "<CONSQ v=" quoted-string [xml:lang= LanguageID ] "/>"
```

`xml:lang` and its value type `LanguageID` are defined in the [XML] specification.

4.4.3 The `PURPOSE` element

`<PURPOSE>` (`xmlns="http://www.w3.org/TR/WD-P3P/vocab`")

purposes for data processing relevant to the Web.

`v` **(*mandatory attribute*)**

the corresponding value

Multiple values can be indicated using multiple `PURPOSE` elements, each with one of the values.

```
[15]purpose     = "<VOC:PURPOSE v=" `"` purposevalue `"` "/>"
```

The values of [16] are provided for reference only. The normative specification is in the harmonized vocabulary [HARMV] document.

```
[16]purposevalue= "current" | ; Completion and Support of Current Activity
                  "admin"   | ; Web Site and System Administration
                  "custom"  | ; Customization of the Site to Individuals
                  "develop" | ; Research and Development
                  "contact" | ; Contacting Visitors for Marketing of Services or Products
                  "other" [" (" string ")"] ; Other Uses
```

4.4.4 The `RECPNT` element

`<VOC:RECPNT>` (`xmlns="http://www.w3.org/TR/WD-P3P/vocab`")

the organizational area, or domain, beyond the service provider and its agents where data may be distributed.

`v` **(*mandatory attribute*)**

the corresponding value

Multiple values can be indicated using multiple `RECPNT` elements, each with one of the values.

```
[16]recipient     = "<VOC:RECPNT v=" `"` recipientvalue `"` "/>"
```

The values of [17] are provided for reference only. The normative specification is in the harmonized vocabulary [HARMV] document.

```
[17]recipientvalue= "ours"  | ; only ourselves and our agents
                    "same"  | ; organizations following our practices
                    "other" | ; organizations following different practices
                    "published" ; unrelated third parties or public forum
```

4.4.5 The REF element

`<REF>` (`xmlns="http://www.w3.org/TR/WD-P3P/basedata"` as default)

describes the data to be transferred or inferred

`name` (***mandatory attribute)***

a string denoting the name of a data element/set. Sets and elements are syntactically distinguished by the presence of a trailing dot after the set name. For example, the trailing dot indicates that `user.Home.` is a data set. Remember that ***names are case-sensitive*** (so, for example, `user.Home.` is different from `USER.HOME.` or `user.home.`). Furthermore, in names ***no underscore symbol ("_") can be present, and no number character can appear immediately following a dot.***

`value`

string denoting the actual value of the data element specified in the `name` attribute. In the case of a data element request, the `value` attribute is missing.

`optional`

"no" or "yes". "no" indicates that the data element is required, while "yes" indicates that the data element is not required. *The default is "no".* The `optional` attribute is used only in proposals. Services may use the optional attribute to indicate that a data request is optional. User agents may send or ignore optional attributes according to the user's preferences. Note that P3P does not include a mechanism for specifying that certain data practices are optional. If services wish to give users a choice of data practices, for example, whether or not data is used for marketing, they may do so by providing multiple alternative proposals that users may choose from.

`source` (***mandatory attribute***)

specifies the mechanism for data transfer

`category` (`xmlns="http://www.w3.org/TR/WD-P3P/vocab"`)

a string denoting the categories of a data element.

The following six attributes are only used when a new (not defined in the P3P [BASEDATA]) data element or set is referenced.

`type`

a string denoting the type of a data element/set.

`typeschema`

a URI denoting the data schema where the type specified in the **`type`** attribute is defined.

`template`

specifies whether or not the corresponding data element is part of a type definition only. If set to 1, this means that that the data element is a type definition, and is not actually representing a data element with an associated value. The default value is 0.

`short`

a string denoting the short display name of the data element/set, no more than 127 [UTF-8] characters.

`long`

a string denoting a longer description of a data element/set, no more than 1023 [UTF-8] characters.

`size`

denotes the maximum number of [UTF-8] characters that are needed to store the data element. *The default value of 0* indicates that the data element can be arbitrarily large.

In addition, the namespace of `REF` can be overridden, using for example an in-line **`xmlns`** attribute, with an URI denoting the name of the data schema the data element/set specified in name belongs to. The default namespace of `REF` is declared in the `PROP` element, but it can be overridden using a local namespace declaration in a `DATA:REF` element.

```
[18] data-reference="<DATA:REF" " name=" quoted-string
                  " source=" `"` sourcevalue `"`
                  [" optional=" yesno] [" value=" quoted-string]
                  [" type=" quoted-string] [" typeschema=" quoted-string]
                  [" template=" yesno] [" VOC:category=" category]
                  [" short=" quoted-string] [" long=" quoted-string]
                  [" size=" `"` number `"`] ; default is 0 (unlimited size)
                  "/>"

[19] sourcevalue   ="service"       |
                  "agent"         |
                  "matched-form" |
                  "extension"
```

Example: To request the user's home address city, all the elements of the data set `user.business.` and (optionally) all the elements of the data set `user.home.phone` and collect them through a matched form, the service would send the following references inside a P3P proposal:

```
<DATA:REF name="user.home.city" source="matched-form"/>
<DATA:REF name="user.home.phone." optional="yes" source="matched-form"/>
<DATA:REF name="user.business." source="matched-form"/>
```

If the user agrees to returning the city and business information and only the international phone code and local area code of her home phone number, she returns the following inside the `txd` tag:

```
<DATA:REF name="user.home.city" value="Cambridge"/>
<DATA:REF name="user.home.phone.intcode" value="1"/>
<DATA:REF name="user.home.phone.loccode" value="617"/>
<DATA:REF name="user.business.street" value="254 Windsor St."/>
<DATA:REF name="user.business.city" value="Cambridge"/>
```
... (*the other values of* `user.business.`) ...

4.4.6 Null Values

In some cases, only some of the values in a data set could have a value (for example, the user could not have a home page, or a fax, or a middle name, etc.). That is to say, for some elements there can be no defined value at all in the user's repository: in such case, elements are said to have a *null value*. The way a client signals that an element has a null value is simply by omitting the corresponding DATA:REF reference. For example, if a client agrees to provide all the user's name information (`user.name.`), and then sends

```
<DATA:REF name="user.name.first." value="Sheila"/>
<DATA:REF name="user.name.last" value="Doherty"/>
<DATA:REF name="user.name.formatted" value="Sheila Doherty"/>
```

that means that in the user's repository all the other fields (prefix, suffix, middle name, nickname) are not defined.

4.4.7 The `source` attribute

The `source` attribute specifies a mechanism for transferring data from the user agent to the service. We define three standard mechanisms, and an extensibility mechanism that allows new data transfer mechanisms to be defined in the future. The standard mechanisms are as follows:

service

The service solicits and collects the data using a method that requires no active involvement of the P3P user agent (for example, standard HTML forms, data collected from standard HTTP header stream, inferred data, etc.). Thus the service maintains full control over the presentation of any prompts to the user. However, the service may not take advantage of the user data repository.

agent

The P3P agent must prompt the user for data and/or retrieve data from the user's data repository. The agent must transfer the data to the service using the HTTP header extension mechanism. The service has no control over how the user will be prompted for the data if it is not available in the repository. User agents should incorporate the short and/or long display names into their prompts rather than using the element name.

matched-form

The service solicits data through a form in the associated HTML content in which the names of the form fields match the data references in the proposal. ***The form field names should use an underscore ("_") notation instead of the normal P3P dot (".")***

notation to separate different components of the attribute name (allowing interoperability with client-side javascript). For example, `user.name.first` in a P3P proposal would be referenced as `user_name_first` in an HTML form. Thus agents may optionally fill out the form with data from the user's repository. When using this method the service maintains full control over the presentation of user prompts, but may take advantage of the user data repository if the user agent has auto-fill support. If the user agent does not have this support (or if none of the requested data is stored in the repository), a standard HTML form is displayed.

extension

A service may also wish to transfer data using another mechanism not defined here, such as an electronic commerce wallet. In that case, the service should specify `source="extension"` to indicate that data will be collected using that mechanism.

`<REF>` elements should always include a source attribute with exactly one value., except when they occur in a data schema definition. In that case the attribute may be left out entirely, it may take one value, or it may take multiple values (the latter case is obtained by multiple `<REF>` elements referencing the same data but with different `source`'s).

4.5 Categories

Categories are attributes of data elements that provide hints to users and user agents as to the intended uses of the data. Categories are vital to making P3P user agents easier to implement and use; they allow users to express more generalized preferences and rules over the exchange of their data. Categories are often included when defining a new element or when referring to data that the user is prompted to type in (as opposed to data stored in the user data repository).

In the current version of P3P, the following tokens are used to denote data categories:

```
[20] category="physical"    | ; Physical Contact Information
              "online"      | ; Online Contact Information
              "uniqueid"    | ; Unique Identifiers
              "financial"   | ; Financial Account Identifiers
              "computer"    | ; Computer Information
              "navigation"  | ; Navigation and Click-stream Data
              "interactive" | ; Interactive Data
              "pref"        | ; Preference Data
              "demograph"   | ; Demographic and SocioEconomic Data
              "content"       ; Content
```

While we specify the category attribute for reference purposes, the normative definitions and values, as well as a more detailed explanation of each category, can be found in the P3P Harmonized Vocabulary Specification [HARMV].

P3P uses *categories* to give users and user agents additional hints as to what type of information is requested from a services. While most data in the Base Data Set is in a known category (or a set of known categories), some data elements can be in a number of different categories, depending on the situation. The former are called *fixed-category data elements* (or "fixed data elements" for short), the latter *variable-category data elements* ("variable data elements"). We will briefly describe both types of elements in the two sections below.

4.5.1 Fixed-Category Data Elements

Most of the elements in the base data set are so called *"fixed"* data elements: they belong into one or at most two category classes. By assigning a category invariably to elements in the base data set, services and users are able to refer to entire groups of elements simply by referencing the corresponding category. For example, using APPEL, the privacy preferences exchange language, users can write rules that prevent their user agent from giving out any data element in a certain category.

When creating new data schemas (see the next section "Creating New Data Sets") for fixed data elements, schema creators have to explicitly enumerate the categories that these element belong to. For example:

```
<DATA:REF name="postal.street.line1"     type="text"
          short="Street Address, Line 1" VOC:category="physical" template="yes"/>
```

In case of multiple categories, multiple elements referencing the same data can be used, each with a different category). For example, in case we want to declare that the data elements in user.name. have both category "physical" and "demograph" we can use:

```
<DATA:REF name="user.name."     type="personname."
          short="User's Name" VOC:category="physical" template="yes" />

<DATA:REF name="user.name."     type="personname."
          short="User's Name" VOC:category="demograph" template="yes" />
```

Please note that the category classes of fixed data elements can **not** be overridden, for example by writing rules or proposals that assign a different category to a known fixed base data element. User Agents MUST ignore such categories and instead use the original category (or set of categories) listed in the schema definition. User Agents CAN preferably alert the user that a fixed data element is used together with a non-standard category class.

4.5.2 Variable-Category Data Elements

Not all data elements in the base data set belong to a pre-determined category class. Some elements can contain information from a range of categories, depending on a particular situation. Such elements are called *variable-category data elements* (or "variable data element" for short). Although most variable data elements in the P3P Base Data Set are combined in the **dynamic.** element set, they can appear in any data set, even mixed with *fixed-category data elements*.

When creating a schema definition for such elements, schema authors MUST NOT list an explicit category attribute, otherwise the element becomes a *fixed* data element. For example when specifying the "Year" *data type*, which can take various categories depending on the situation (e.g. when used for a credit card expiration date vs. for a birth date), the following schema definition is used:

```
<DATA:REF name="date.year" type="number" size="6"
          short="Year"     template="yes"/>  <!-- Variable Data Element -->
```

This allows new schema extensions that reference such variable-category *data types* to assign a specific category to derived elements, depending on their usage in that extension. For example, an E-commerce schema extension could thus define a credit card expiration date as follows:

```
<DATA:REF name="Card.ExpDate."         type="date."
          short="Card Expiration Date" VOC:category="financial" template="yes"/>
```

Under these conditions, the variable data type **date.** is assigned a fixed category Financial Account Identifiers when being used for specifiying a credit card expiration date. Please see the next section "Creating New Data Sets" for more information.

Note that while user preferences can list such variable data elements without any additional category information (effectively expressing preferences over *any* usage of this element), services MUST always explicitly specify the categories that apply to the usage of a variable data element in their particular proposal. This information has to appear as an attribute to the corresponding `DATA:REF` element listed in the proposal, for example as in:

```
<P3P:PROP>
   ...
   <DATA:REF name="dynamic.cookies" VOC:category="uniqueid">
   ...
</P3P:PROP>
```

where a service declares that cookies are used for identifying the user at this site (i.e. category Unique Identifiers).

4.6 Creating New Data Sets

Services may declare and use new data elements by creating a data schema and referencing it in a proposal using the `DATA` namespace (`xmlns:DATA`). When data elements beyond those defined in the base data set [BASEDATA] are referenced in a proposal, the `DATA:REF` element must include values for the following attributes: `name, category`, `type`, `typeschema`, and `short` (`typeschema` may be omitted if it has the same value as the `DATA` namespace)`.` If the data schema URI does not domain match the origin server URI, user agents MUST check the data schema to verify the consistency of the information. Otherwise the information in the main data schema MAY be checked in order to verify consistency of the information. In case of a mismatch, either an ERR-IF reason code (Invalid Format) or the generic ERR code must be returned. If for any other reason the user is unable to reconstruct the needed information, either a SRY-DU reason code (Data Unavailable) or a generic SRY code must be returned.

The format of a new data schema is a special proposal of the form

```
<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax"
 xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
 xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
<USES><STATEMENT>
  ....
</STATEMENT></USES></PROP>
```

A data block is enclosed within the `<STATEMENT>` tag and contains references to the new data elements. References can be made using the `<DATA:REF>` tag and the following attributes: `name, type, typeschema, template, VOC:category, short, long, size, source.`

For every data element, all of the information except `long` and `source` are mandatory. If `source` is present, it indicates that the element must only be collected using the specified data transport mechanism. If `source` is missing, there are no restrictions placed on how the element may be collected. If any attribute is missing, it is presumed to be present with an empty string. In the case of the `typeschema`, the empty string value has the special meaning that the type schema coincides with the namespace of the corresponding `REF` element.

For example, suppose the company HyperSpeed wants to build the following data schema:

`car.model`
`car.color`
`car.built.year`
`car.built.where.` (of basic type `postal.`)
`car.price`
`bike.model`
`bike.color`
`bike.built.year`
`bike.built.where.` (of basic type `postal.`)
`bike.price`

Then, it could place the following code at http://www.HyperSpeed.com/models-schema

```
<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax"
 xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
 xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
<USES><STATEMENT VOC:id="nonid">
 <VOC:RECPNT v="ours"/>
 <VOC:PURPOSE v="contact"/>
 <DATA:REF name="car.model" type="text" short="Model"
   VOC:category="pref" size="63"/>
 <DATA:REF name="car.color" type="text" short="Color"
   VOC:category="pref" size="63"/>
 <DATA:REF name="car.built.year" type="number" short="ConstructionYear"
   VOC:category="pref" size="63"/>
 <DATA:REF name="car.built.where." type="postal."  short="Construction Place"
   VOC:category="pref" size="63"/>
 <DATA:REF name="bike." type="car."
   typeschema="http://www.HyperSpeed.com/models-schema"/>
</USES></STATEMENT></PROP>
```

Note that *every time a data set is created, it can be implicitly used as a type*, just like the `car.` case above. However, in some situations one may wish to define a type without creating a specific element within the user's repository. This can be accomplished by using the `template` attribute in the `<DATA:REF>` . Setting the value to yes, `template="yes"` (default is "no"), means that the corresponding data element is part of a type definition only, and is not actually representing a data element with an associated value. For example, HyperSpeed might want to define a `GenericModel.` type of general utility, and then instantiating it with `car.` and `bike.` This could be done with the following schema:

```
<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax"
 xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
 xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
<USES><STATEMENT VOC:id="nonid">
 <VOC:RECPNT v="ours"/>
 <VOC:PURPOSE v="contact"/>
 <DATA:REF name="GenericModel.model" type="text" short="Model" template="yes"
   VOC:category="7" size="63"/>
 <DATA:REF name="GenericModel.color" type="text" short="Color" template="yes"
   VOC:category="7" size="63"/>
 <DATA:REF name="GenericModel.built.year" type="number" short="Construction Year"
   template="yes" VOC:category="7" size="63"/>
 <DATA:REF name="GenericModel.built.where." type="postal."
   short="Construction Place" template="yes" VOC:category="7" size="63"/>
 <DATA:REF name="car." type="GenericModel."
   typeschema="http://www.HyperSpeed.com/models-schema" short="Car"/>
 <DATA:REF name="bike." type="GenericModel."
   typeschema="http://www.HyperSpeed.com/models-schema" short="Bike"/>
```

```
</USES></STATEMENT></PROP>
```

Creators of new schema should be aware that unless they place restrictions on the source attribute (i.e. `source="service"` or use an extension that does not store data in the user data repository), user agents may automatically store elements from the new schema in the user data repository after the user types them in. Some implementations may include user data repositories that do not use security measures appropriate for storing sensitive data.

In order to provide *multilingual support* for data schema files, a server can supply the right alternative based on the HTTP `Accept-Language` header.

P3P allows users to store frequently requested information, such as user name and postal address, in a client-side repository. While most data in the P3P Base Data Set [BASEDATA] (as well the majority of schema extensions) will most likely be stored in the repository, P3P also supports the use of *non-repository data*. Such data is either information that is never stored in a repository and has to be manually entered by the user every time the element is requested, or data that constitutes information which is dynamically created by the user agent or operating system, such as clickstream information or the browser identification string.

When creating new schema definitions, schema designers can either specify that user agents should try to create corresponding fields in the user's repository (thus facilitating repeated transmission of the information), or that the element is dynamic and should not be included in the repository. The following subsections explain those two concepts in more detail.

4.6.1 Repository Data

Any element in a schema definition is assumed to be part of the user's repository. Thus, a standard schema definition of a (fixed) repository element would look like this:

```
    <DATA:REF name="personname.first" type="text"
              short="First Name"      VOC:category="physical" template="yes"/>
```

However, even though an element is part of the user repository, a P3P proposal could explicitly specify that it wants the user to manually enter it (for example through an HTML form), ignoring any available data in the repository:

```
    <P3P:PROP>
       ...
       <DATA:REF name="user.name.first" source="service">
       ...
    </P3P:PROP>
```

Applications can of course still offer a repository-based auto-fill functionality, even though the site did not request any repository data. Also note that it is up to user agent implementations to decide what action to take when encountering a new repository-stored data schema. Some implementation could for example automatically add the element to the user's repository (with an empty value), others might want to prompt the user for a decision.

4.6.2 Non-Repository Data

As explained before, attribute values set in data schema definitions can not be overridden. Thus, in order to force a certain data element to always be *outside* of the user repository, schema designers simply have to set the `source` attribute to `service`:

```
    <DATA:REF name="dynamic.clickstream.client"
              short="Click-stream collected on the client"
                             type="boolean" source="service"
                              VOC:category="navigation" template="yes"/>
```

The above example ensures that `dynamic.clickstream.server` will never become part of the user's repository, but is instead either manually collected (for example through HTML forms) or implicitly transmitted as part of the connection (which is the case for clickstream data). In either case, a P3P user agent could (depending on the user's preferences) inform the user about the collection of this data element and its terms, but would take no further action for transmitting the data.

5. Appendices

Appendix 1: References (Normative)

[APPEL]

M. Langheinrich (Ed.). "A P3P Preference Exchange Language (APPEL)" World Wide Web Consortium.

[BASEDATA]

M. Marchiori (Ed.). "P3P Base Data Set." World Wide Web Consortium, Working Draft.

[DSIG]

Y. Chu, P. DesAutels, B. LaMacchia, P. Lipp. "PICS Signed Labels (DSig) 1.0 Specification," World Wide Web Consortium Recommendation 27 May 1998.

[HARMV]

J. Reagle (Ed.). "P3P Harmonized Vocabulary Specification," World Wide Web Consortium, Working Draft.

[HTTP1.0]

T. Berners-Lee, R. Fielding, H. Frystyk Nielsen, "RFC1945 -- Hypertext Transfer Protocol -- HTTP/1.0," W3C/MIT, UC Irvine, W3C/MIT, May 1996.

[HTTP1.1]

R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, T. Berners-Lee, "RFC2068 -- Hypertext Transfer Protocol -- HTTP/1.1," UC Irvine, Digital Equipment Corporation, MIT.

[HTTP-EXT]

H. Frystyk, P. Leach, S. Lawrence. "HTTP Extensions" (draft-frystyk-http-extensions-03.txt). Jan 1999. IETF Internet Draft.

[ISO3166]

"ISO3166: Codes for The Representation of Names of Countries." International Organization for Standardization.

[KEY]

S. Bradner. "RFC2119 -- Key words for use in RFCs to Indicate Requirement Levels." March 1997.

[MD5]

R. Rivest. "RFC 1321 -- The MD5 Message-Digest Algorithm," MIT. April 1992.

[MIME]

N. Freed, N. Borenstein. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." November 1996.

[RDF]

O. Lassila, R. Swick (Eds.). "Resource Description Framework (RDF) Model and Syntax Specification." World Wide Web Consortium Recommendation. 22 February 1999.

[SSL]

A. Freier, P. Karlton, P. Kocher. "SSL 3.0 Specification." (http://home.netscape.com/eng/ssl3/index.html)

[STATE]

Network Working Group, D. Kristol, Bell Laboratories, Lucent Technologies; Category: Standards Track HTTP State Management Mechanism. (ftp://ftp.isi.edu/in-notes/rfc2109.txt)

[URI]

T. Berners-Lee, R. Fielding, and L. Masinter. "Uniform Resource Identifiers (URI): Generic Syntax and Semantics." August 1998. RFC 2396. [Updates RFC1738]

[UTF-8]

F. Yergeau. "RFC2279 -- UTF-8, a transformation format of ISO 10646." January 1998.

[VCARD]

"vCard - The Electronic Business Card Version 2.1." Internet Mail Consortium, September 18, 1996.

[XML]

T. Bray, J. Paoli, C. M. Sperberg-McQueen. "Extensible Markup Language (XML) 1.0 Specification." World Wide Web Consortium, Recommendation. 10 February 1998.

[**XML-Data**]

A. Layman *et al.* "XML-Data." World Wide Web Consortium, Note. 05-January-1998.

[**XML-Name**]

T. Bray, D. Hollander, A. Layman. "Namespaces in XML." World Wide Web Consortium, Recommendation, 14-January-1999.

Appendix 2: ABNF Notation (Non-normative)

The formal grammar of P3P is given in this specification using a slight modification of the ABNF defined in http://info.internet.isi.edu/in-notes/rfc/files/rfc2234.txt . The following is a simple description of the ABNF.

`name = (element)`

where <name> is the name of the rule, <elements> is one or more rule names or terminals combined through the operands provided below. Rule names are case-insensitive.

`(element1 element2)`

elements enclosed in parentheses are treated as a single element, whose contents are strictly ordered.

`<a>*<b>element`

at least <a> and at most <b> occurrences of the element.

*(1*4<element> means one to four elements.)*

`<a>element`

exactly <a> occurrences of the element.

(4<element> means exactly 4 elements.)

`<a>*element`

<a> or more elements

(4<element> means 4 or more elements.)*

`*<b>element`

0 to <b> elements.

*(*5<element> means 0 to 5 elements.)*

`*element`

0 or more elements.

(<element> means 0 to infinite elements.)*

`[element]`

optional element, equivalent to *1(element).

([element] means 0 or 1 element.)

`"string"` or **`'string'`**

matches a literal string matching that given inside the double quotes.

Other notations used in the productions are:

; or **`/* ... */`**

comment.

Appendix 3: Working Group Contributors

The following individuals participated in the P3P Specification Working Group:

[fill in later]

The P3P Specification Working Group inherited most of the specification from previous P3P working groups. The Working Group would like to acknowledge the contributions of the members of these previous groups:

[fill in later]

P3P Base Data Set

W3C Working Draft 26 August 1999

This Version
http://www.w3.org/TR/1999/WD-P3P-19990826/basedata

Latest Public Version:
http://www.w3.org/TR/WD-P3P/basedata

Previous Version:
http://www.w3.org/TR/1999/WD-P3P-19990407/basedata.html

Editor:
Massimo Marchiori, W3C, (massimo@w3.org)

Abstract

This document, part of the P3P specification, specifies the names of base P3P data elements, sets, and their data types.

Status of This Document

This is a subspecification of the P3P specification for review by W3C members and other interested parties. This document has been produced as part of the P3P Activity, and will eventually be advanced toward W3C Recommendation status. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress." The underlying concepts of the draft are fairly stable and we encourage the development of experimental implementations and prototypes so as to provide feedback on the specification. However, this Working Group will not allow early implementations to affect their ability to make changes to future versions of this document.

This draft document will be considered by W3C and its members according to W3C process. This document is made public for the purpose of receiving comments that inform the W3C membership and staff on issues likely to affect the implementation, acceptance, and adoption of P3P.

Send comments to www-p3p-public-comments@w3.org (archived at http://lists.w3.org/Archives/Public/www-p3p-public-comments/).

Table of Contents

1. Introduction

P3P uses a set of Base Data Elements to provide a common platform for services and user agents to request and exchange information. All P3P-compliant user agent implementations MUST be aware of these data elements. The P3P Base Data set includes two element sets, **user.** and **dynamic.**. The user. set includes elements that users might provide values for, while the dynamic. set includes elements that are dynamically generated in the course of a user's browsing session. User agents may support a variety of mechanisms that allow users to provide values for the elements in the user. set, including mechanisms that support multiple personae. Users may choose not to provide values for these data elements.

Data Elements can be classified along two axes: whether or not they are in a fixed category (using the category attribute), and whether or not they are part of the user's repository (using the source attribute.) Schema Designers can use these attributes within their schema definitions to define an invariable category and/or source value for each element. Once defined, these values cannot be changed when referencing such elements from within user preferences, P3P proposals, or other schema definitions.

However, if left undefined, those attributes MUST be explicitly listed in each P3P proposal referencing such elements. Users can have different preferences depending on different attribute-values for the same element. And in the case of undefined attributes within data *types*, other schema definition can explicitly set categories and/or source information in derived elements (otherwise the original definition overrides any value in the derived schema).

Please refer to the section Creating New Data Sets in the P3P Syntax Specification for more information on the usage of these two attributes.

Finally, remember that, as specified in the syntax spec, when soliciting data via an HTML form the form field names should instead use an underscore ("_") to separate the different levels of the attribute name (e.g. `user.name.First` must be referenced as `User_Name_First`). This allows interoperability with client-side javascript which also uses the dot notation to access form field names and values.

2. Required Base Data Elements and Sets

The following are the base data elements and sets. The members of this working group expect that in the future, *there will be demand for the creation of other data sets and elements.* Obvious applications include catalogue, payment, and agent/system attribute schemas. (An extensive set of system elements is provided in a white paper draft (W3C Members only) and are based on http://www.w3.org/TR/NOTE-agent-attributes.)

Each table below specifies a **set**, the elements within the set, the category associated with the element, its type, and the display name shown to users. More than one category may be associated with a fixed data element. However, we have tried to assign each base data element to only one category whenever possible. We recommend that data schema designers do the same.

Note that variable data elements list a wildcard "*" in the category field, indicating that services must explicitly choose one or more categories for this element should they include it in a proposal. They also feature a different background color.

2.1 User Data

The **user.** data set includes general information about the user.

user.	**Category**	**Type**	**Short display name**
name.	Physical Contact Information, Demographic and SocioEconomic Data	personname.	User's Name
bdate.	Demographic and SocioEconomic Data	date.	User's Birth Date
cert.	Unique Identifiers	certificate	User's Identity certificate
gender	Demographic and SocioEconomic Data	gender	User's gender
employer	Demographic and SocioEconomic Data	text	User's Employer
department	Demographic and SocioEconomic Data	text	Department or division of organization where user is employed
jobtitle	Demographic and SocioEconomic Data	text	User's Job Title
home.	Physical Contact Information, Online Contact Information, Demographic and SocioEconomic Data	contact.	User's Home Contact Information
business.	Physical Contact Information, Online Contact Information, Demographic and SocioEconomic Data	contact.	User's Business Contact Information

Note, that this data set includes elements that are actually sets of data themselves. These sets are defined in the data types subsection of this document. The short display name for an individual element contained within a data set is defined as the concatenation of the short display names that have been defined for the set and the element, separated by commas. For example, the short display name for `user.home.postal.Postalcode` would be "User's Home Contact Information, Postal Address Information, Postal code". User agent implementations may prefer to develop their own short display names rather than using the concatenated names when prompting users for information.

2.2 Dynamic Data

In some cases, there is a need to specify data elements that do not have associated data values in the user's repository (e.g., when a service wishes to declare it keeps HTTP logs): these are the so-called *non-repository* elements explained in the P3P Syntax Specification. In the P3P Base Data Set, all such non-repository elements are grouped under the **dynamic.** data set. While elements from other data sets (such as **user.**) might require the user agent to perform steps to retrieve such information from the user repository, elements from the **dynamic.** data set are never to be found in the repository but instead are collected *outside* the scope of P3P, for example through HTTP headers, HTML forms, or implicitly on the server side by collecting clickstream logs.

dynamic.	Category	Type	Short display name
clickstream.client	Navigation and Click-stream Data	boolean	Click-stream collected on the client
clickstream.server	Navigation and Click-stream Data	boolean	Click-stream collected on the server
cookies	*	boolean	cookies are processed (read/write)
http.useragent	Computer Information	text	User Agent information
http.referrer	Navigation and Click-stream Data	uri	Last URI requested by the user
miscdata	*	text	Miscellaneous non-base data set information
searchtext	Interactive Data	text	Search terms
interactionrecord	Interactive Data	boolean	server stores the transaction history

These elements are often implicit in navigation or Web interactions. They should be used with categories to describe the type of information collected through these methods. A brief description of each element follows.

"clickstream.client" should be used when the server accesses off-line browsing information that has been collected by the user's client. Some versions (e.g. 5.0) of Microsoft's Internet Explorer are known to support such behavior.

"clickstream.server" will probably apply to almost all sites on the Web today. It must be used whenever page access data is kept on the server side. Almost all known Web server implementations today will by default create such an access log, often including origin of the request (IP address or DNS name), time, requested resource, HTTP answer code and transferred bytes. Any combination of resource name and originating address should be considered *clickstream* data (i.e. it allows the reconstruction of a visitors movements through the site) and should be declared.

Please note that the logging of referer or user agent information (included in the headers of the HTTP request by many browsers) should explicitly by declared using the http.useragent and http.referrer data elements.

"cookies" should be used whenever information is placed on a user's machine using the HTTP cookie mechanism in order to be "solicited" (i.e. automatically sent) later. Please note that "cookies" is a *variable data element* and requires the explicit declaration of usage categories in a proposal.

"http.useragent" indicates that the server stores additional information about the user agent in its logs, such as operating system, browser software and version.

"http.referrer" indicates that the server stores additional information about the page the user viewed previously, as indicated by the HTTP_REFERER header (the HTTP spec uses two "R" instead of three!).

The "miscdata" element references information collected by the service that is not described by any element in the currently available (base) data element sets. Sites MUST reference a separate miscdata element in their proposals for each category of miscdata they collect.

"searchtext" is a specific type of solicitation used for searching and indexing sites. For example, if the only fields on a search engine page are search fields, the site only needs to disclose that data element.

The "interactionrecord" element should be used if the server is keeping track of the interaction it has with the user (i.e. information other than clickstream data, for example account transactions, etc). This element is only meant to inform the user that such information will be retained, but does not indicate how long such data will be kept.

Proposals that contain one or more of the *Variable Data Elements* above explicitly declare the category (as defined by the Vocabulary) of the information they solicit, for example:

```
    <P3P:PROP>
       ...
       <DATA:REF name="dynamic.miscdata" VOC:category="1">
```

```
     ...
  </P3P:PROP>
```

when asking (through a form) for a user's IRC name (which would be in category 1 Online Contact Information).

Please note that if a form asks for elements from a known schema set -- for example the user's name or postal address -- these SHOULD be *explicitly* listed in the proposal using the element's name and the "`source="service"`" attribute, instead of referencing them indirectly through a "dynamic.miscdata" element.

3. Data Types

3.1 Dates

The **date.** type is a structured type that specifies a date. Since date information can be used in different ways, depending on the context, all **date.** information is tagged as being of "variable" category. Schema definitions have to explicitly set the corresponding category in the element referencing this data type. For example, soliciting the birthday of a user might be "Demographic and SocioEconomic Data", while the expiration date of a credit card belongs to the "Financial Account Identifiers" category.

date.	**Category**	**Type**	**Short display name**
year	*	number	Year
month	*	number	Month
day	*	number	Day
hour	*	number	Hour
minute	*	number	Minute
second	*	number	Second
fractionsecond	*	number	Fraction of Second
timezone	*	text	Time Zone

All the fields in the **date.** type correspond to those in the most informative profile of the time standard ISO8601.

3.2 Names

The **personname.** type is a structured type that specifies information about the naming of a person.

personname.	Category	Type	Short display name
prefix	Demographic and SocioEconomic Data	text	Name Prefix
first	Physical Contact Information	text	First Name
middle	Physical Contact Information	text	Middle Name
last	Physical Contact Information	text	Last Name
suffix	Demographic and SocioEconomic Data	text	Name Suffix
formatted	Physical Contact Information, Demographic and SocioEconomic Data	text	formatted Name
nickname	Demographic and SocioEconomic Data	text	Nickname

3.3 certificates

The **certificate.** type is a structured type to specify identity certificates (like, for example, X.509).

certificate.	Category	Type	Short display name
key	Unique Identifiers	binary	Certificate Key
format	Unique Identifiers	text	Certificate Format

The "format" field is an IANA registered public key or authentication certificate format, while the "key" field contains the corresponding certificate key.

3.4 Telephones

The **phonenum.** type is a structured type that specifies the characteristics of a phone number.

phonenum.	Category	Type	Short display name
intcode	Physical Contact Information	number	International Phone code
loccode	Physical Contact Information	number	Local Phone Area code
number	Physical Contact Information	number	Phone Number
ext	Physical Contact Information	number	Phone Extension

comment	Physical Contact Information	text	Phone Optional Comments

3.5 Contact Information

The **contact.** type is a structured, redirected, type to other types. This is done so that services can specify precisely which set of data they need.

contact.	**Category**	**Type**	**Short display name**
postal.	Physical Contact Information, Demographic and SocioEconomic Data	postal.	Postal Address Information
telecom.	Physical Contact Information	telecom.	Telecommunications Information
online.	Online Contact Information	online.	Online Address Information

3.5.1 Postal

The **postal.** type is a structured type that specifies a postal mailing address.

postal.	**Category**	**Type**	**Short display name**
name.	Physical Contact Information, Demographic and SocioEconomic Data	personname.	Name
street.line1	Physical Contact Information	text	Street Address 1
street.line2	Physical Contact Information	text	Street Address 2
street.line3	Physical Contact Information	text	Street Address 3
city	Physical Contact Information	text	City
stateprov	Physical Contact Information	text	State or Province
postalcode	Demographic and SocioEconomic Data	text	Postal code
countrycode	Demographic and SocioEconomic Data	Country	Country code
country	Demographic and SocioEconomic Data	text	Country Name
organization	Physical Contact Information, Demographic and SocioEconomic Data	text	Organization Name
formatted	Demographic and SocioEconomic Data	text	formatted Postal Address

Using three distinct fields for the street information allows service providers and user agents to split long addresses into multiple lines during solicitation. However, since all fields share the common **street.** prefix, this shorthand form can be used in both preference files and proposals to reference all three fields at once.

The "formatted" field is used to specify the formatted text corresponding to the delivery address, as it could for example be printed on a label.

3.5.2 Telecommunication

The **telecom.** type is a structured type that specifies telecommunication information about a person.

telecom.	Category	Type	Short display name
phone.	Physical Contact Information	phonenum.	Phone number
fax.	Physical Contact Information	phonenum.	Fax number
mobile.	Physical Contact Information	phonenum.	Mobile Phone number
pager.	Physical Contact Information	phonenum.	Pager number

3.5.3 Online

The **online.** type is a structured type that specifies online information about a person.

online.	Category	Type	Short display name
email	Online Contact Information	text	Email Address
uri	Online Contact Information	uri	Home Page Address

3.6 Primitive Data Types

This specification uses the following primitive data element datatypes:

Primitive DataType	Definition
text	[UTF-8]
gender	"M" or "F".
boolean	"0" or "1".
binary	Base64 per RFC-1531. [MIME]
number	text composed with the digits "0", "1", "2", "3", "4", "5", "6", "7", "8", "9".
Country	[ISO3166]
uri	[URI]

4. The Data Schema

The data schema corresponding to the P3P base data set follows. In order to improve legibility, we have indented and aligned the code along various attribute names.

```
<PROP xmlns="http://www.w3.org/TR/WD-P3P/syntax/"
xmlns:VOC="http://www.w3.org/TR/WD-P3P/vocab"
xmlns:DATA="http://www.w3.org/TR/WD-P3P/basedata">
<USES><STATEMENT>
<!-- ********** Base Data Types ********** -->

<!-- "date." Data Type -->
<DATA:REF name="date.year"
          short="Year"
                          type="number" size="6"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.month"
          short="Month"
                          type="number" size="2"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.day"
          short="Day"
                          type="number" size="2"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.hour"
          short="Hour"
                          type="number" size="2"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.minute"
      ;  short="Minutes"
                          type="number" size="2"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.second"
          short="Second"
                          type="number" size="2"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.fractionsecond"
```

```
          short="Fraction of Second"
                         type="number" size="6"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="date.timezone"
          short="Time Zone"
                          type="text"   size="10"
                          template="yes"/>  <!-- Variable Data
Element -->

<!-- "personname." Data Type -->
<DATA:REF name="personname.Prefix"
          short="Name Prefix"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="personname.first"
          short="First Name"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="personname.middle"
          short="Middle Name"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="personname.last"
          short="Last Name"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="personname.suffix"
          short="Name Suffix"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="personname.formatted"
          short="formatted Name"
                          type="text"
                         VOC:category="physical"
template="yes"/>
<DATA:REF name="personname.formatted"
          short="formatted Name"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="personname.nickname"
          short="Nickname"
                          type="text"
                          VOC:category="demograph"
template="yes"/>

<!-- "certificate." Data Type -->
<DATA:REF name="certificate.key"
          short="Certificate Key"
                          type="binary" size="0"
                          VOC:category="uniqueid"
template="yes"/>
<DATA:REF name="certificate.format"
          short="Certificate format"
                          type="number" size="128"
                          VOC:category="uniqueid"
template="yes"/>

<!-- "phonenum." Data Type -->
<DATA:REF name="phonenum.intcode"
          short="International Phone Code"
                          type="number" size="11"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="phonenum.loccode"
          short="Local Phone Area Code"
                          type="number" size="11"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="phonenum.number"
          short="Phone Number"
                          type="number" size="30"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="phonenum.ext"
          short="Phone Extension"
                          type="number" size="11"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="phonenum.comment"
          short="Phone Optional Comments"
                          type="text"
                          VOC:category="physical"
template="yes"/>

<!-- "contact." Data Type" -->
<DATA:REF name="contact.postal."
          short="Postal Address Information"
                          type="postal."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="contact.postal."
          short="Postal Address Information"
                          type="postal."
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="contact.telecom."
          short="Telecommunications Information"
                          type="telecom."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="contact.online."
          short="Online Address Information"
                          type="online."
                          VOC:category="online" template="yes"/>

<!-- "postal." Data Type -->
<DATA:REF name="postal.name."
          short="Name"
                          type="personname."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.name."
          short="Name"
                          type="personname."
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="postal.street.line1"
          short="Street Address, Line 1"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.street.line2"
          short="Street Address, Line 2"
                          type="text"
                          VOC:category="physical"
```

```
template="yes"/>
<DATA:REF name="postal.street.line3"
          short="Street Address, Line 3"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.city"
          short="City"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.stateprov"
          short="State or Province"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.postalcode"
          short="Postal Code"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="postal.organization"
          short="Organization Name"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.organization"
          short="Oranization Name"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="postal.formatted"
          short="Formatted Postal Address"
                          type="text"
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="postal.formatted"
          short="Formatted Postal Address"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="postal.country"
          short="Country Name"
                          type="text"
                          VOC:category="demograph"
template="yes"/>
<DATA:REF name="postal.countrycode"
          short="Country Code"
                          type="Country" size="2"
                          VOC:category="demograph"
template="yes"/>

<!-- "telecom." Data Type -->
<DATA:REF name="telecom.phone."
          short="Phone Number"
                          type="phonenum."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="telecom.fax."
          short="Fax Number"
                          type="phonenum."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="telecom.mobile."
          short="Mobile Phone Number"
                          type="phonenum."
                          VOC:category="physical"
template="yes"/>
<DATA:REF name="telecom.pager."
          short="Pager Number"
                          type="phonenum."
                          VOC:category="physical"
template="yes"/>

<!-- "online." Data Type -->
<DATA:REF name="online.email"
          short="Email Address"
                          type="text"
                          VOC:category="online" template="yes"/>
<DATA:REF name="online.uri"
          short="Home Page Address"
                          type="uri"
                          VOC:category="online" template="yes"/>

<!-- ********** Base Data Sets ********** -->

<!-- "dynamic." Data Set -->
<DATA:REF name="dynamic.clickstream.client"
          short="Click-stream collected on the client"
```

```
                          type="boolean" source="service"
                          VOC:category="navigation"/>
<DATA:REF name="dynamic.clickstream.server"
          short="Click-stream collected on the server"
                          type="boolean" source="service"
                          VOC:category="navigation"/>
<DATA:REF name="dynamic.cookies"
          short="cookies are processed (read/write)"
                          type="boolean" source="service"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="dynamic.http.useragent"
          short="User Agent information"
                          type="text" source="service"
                          VOC:category="navigation"/>
<DATA:REF name="dynamic.http.referrer"
          short="Last URI requested by the user"
                          type="uri" source="service"
                          VOC:category="navigation"/>
<DATA:REF name="dynamic.miscdata"
          short="Miscellaneous non base data set information"
                          type="text" source="service"
                          template="yes"/>  <!-- Variable Data
Element -->
<DATA:REF name="dynamic.searchtext"
          short="Search terms"
                          type="text" source="service"
                          VOC:category="interactive"/>
<DATA:REF name="dynamic.interactionrecord"
          short="server stores the transaction history"
                          type="boolean" source="service"
                          VOC:category="interactive"/>

<!-- "user." Data Set -->
<DATA:REF name="user.name."
          short="User's Name"
                          type="personname."
                          VOC:category="physical"/>
<DATA:REF name="user.name."
          short="User's Name"
                          type="personname."
                          VOC:category="demograph"/>
<DATA:REF name="user.bdate."
          short="User's Birth Date"
```

```
                          type="date."
                          VOC:category="demographic"/>
<DATA:REF name="user.cert."
          short="User's Identity certificate"
                          type="certificate."
                          VOC:category="uniqueid"/>
<DATA:REF name="user.gender"
          short="User's gender"
                          type="gender"
                          VOC:category="demograph"/>
<DATA:REF name="user.jobtitle"
          short="User's Job Title"
                          type="text"
                          VOC:category="demograph"/>
<DATA:REF name="user.home."
          short="User's Home Contact Information"
                          type="contact."
                          VOC:category="physical"/>
<DATA:REF name="user.business."
          short="User's Business Contact Information"
                          type="contact."
                          VOC:category="physical"/>
<DATA:REF name="user.employer"
          short="Name of User's Employer"
                          type="text"
                          VOC:category="demograph"/>
<DATA:REF name="user.department"
          short="Department or division of organization where
user is employed"
                          type="text"
                          VOC:category="demograph"/>

</STATEMENT></USES>
</PROP>
```

5. Appendix: References (Normative).

[**HTTP1.1**]

R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, T. Berners-Lee, "RFC2068 -- Hypertext Transfer Protocol -- HTTP/1.1," UC Irvine, Digital Equipment Corporation, MIT.

[**ISO3166**]

"ISO3166: Codes for The Representation of Names of Countries." International

Organization for Standardization.

[**MIME**]

N. Freed, N. Borenstein. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." November 1996.

[SYNTAX]

M. Marchiori (Ed.). "P3P Syntax Specification." World Wide Web Consortium, Working Draft.

[**URI**]

T. Berners-Lee, R. Fielding, and L. Masinter. "Uniform Resource Identifiers (URI): Generic Syntax and Semantics." August 1998. RFC 2396. [Updates RFC1738]

[UTF-8]

F. Yergeau. "RFC2279 -- UTF-8, a transformation format of ISO 10646." January 1998.

[**VCARD**]

"vCard - The Electronic Business Card Version 2.1." Internet Mail Consortium, September 18, 1996.

[**XML**]

T. Bray, J. Paoli, C. M. Sperberg-McQueen. "Extensible Markup Language (XML) 1.0 Specification." World Wide Web Consortium, Recommendation. 10 February 1998.

[**XML-Data**]

A. Layman *et al.* "XML-Data." World Wide Web Consortium, Note. 05-January-1998.

6. Acknowledgements

[fill in later]

P3P Harmonized Vocabulary

W3C Working Draft 26 August 1999

This Version

http://www.w3.org/TR/1999/WD-P3P-19990826/vocab

Latest Version:

http://www.w3.org/TR/WD-P3P/vocab.html

Previous Version:

http://www.w3.org/TR/1999/WD-P3P-19990407/vocab

Editor:

Joseph Reagle, W3C, (mailto:reagle@w3.org)

Abstract

This document of the P3P specification specifies the English language semantics for privacy related disclosures such as categories, purpose, identifiable use, recipients, and access.

Status of This Document

This is a subspecification of the P3P specification for review by W3C members and other interested parties. This document has been produced by the P3P Harmonized Vocabulary WG as part of the P3P Activity. While this document will eventually be advanced toward W3C Recommendation status, it is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress." The underlying concepts of the draft are fairly stable and we encourage the development of experimental implementations and prototypes so as to provide feedback on the specification. However, this Working Group will not allow early implementations to affect their ability to make changes to future versions of this document.

This draft document will be considered by W3C and its members according to W3C process.

This document is made public for the purpose of receiving comments that inform the W3C membership and staff on issues likely to affect the implementation, acceptance, and adoption of P3P.

Send comments to www-p3p-public-comments@w3.org (archived at http://lists.w3.org/Archives/Public/www-p3p-public-comments/).

Table of Contents

1 Introduction

The P3P Specification [P3P] specifies an [XML] / [RDF] application that defines the structure, or grammar, of a P3P **proposal**. This document, the **harmonized vocabulary**, describes the terms that fit into the P3P grammar; this process is technically called the "semantic definition of an XML/RDF schema or vocabulary." For example, the P3P specification states that P3P **statements** must declare the purposes for which data are collected, this document specifies a list of six such purposes and their meaning.

P3P can support multiple schemas. However, P3P is likely to be most effective when a single base vocabulary is widely used since information practice statements are most useful when they can be readily understood by users and their computer agents. *Complementary* vocabularies may develop to cater to jurisdiction-specific concerns not addressed by the base vocabulary. This can be easily accomplished through the XML-namespace [XML-names] facility, which allows tags from different XML schemas to be intermixed. However, the semantics of this specification always dominate those of an external namespace. For instance, someone cannot place an attribute within a proposal that says "and this proposal is void on Tuesdays" and argue this excuses them from the semantics defined in the P3P specification.

Therefore, this document includes a base set of vocabulary elements useful for expressing privacy policies reflective of a diversity of privacy laws, self-regulatory norms, and cultural notions about privacy. This vocabulary can be used to express policies as diverse as anonymous browsing to the provision of personalized Web content and services. However, P3P implementations need not restrict themselves solely to vocabularies defined within this document.

Note, in addition to the terms specified in the harmonized vocabulary, P3P requires services to specify in their proposals the service provider's identity (**entity**), an experience space to which their practices apply (e.g., **realm**: http://www.w3.org), the location at which users can find a human-readable explanation of the service's privacy policies (**discURI**) and an optional human-readable description of the result (e.g., **consequence**: "to offer customized sports updates"). In addition, services may specify an "assuring party" that attests that the service provider will abide by its proposal (**assurance**), follow guidelines in the processing of data, or other relevant assertions. Entity, realm, discURI, consequence, and assurance elements are fully specified in the P3P Syntax Specification.

Security issues and protocols are not addressed by this document. Information about the characteristics and strength of those protocols is critical to a user's decision regarding the transmission of information. However, an assumption of P3P is that communication and storage security is achieved through means other than P3P itself (such as SSL).

Legal issues regarding law enforcement demands for information are not addressed by this document. It is possible that a service provider that otherwise abides by its proposal of not redistributing data to others may be required to do so by force of law.

In this document we introduce the specific terms of the harmonized vocabulary along with their defintions. The words in square brackets next to each term designate an abbreviated version of the term used in XML/RDF representations.

Comment: Much of the work done on this schema was conducted under significant time pressure. Accordingly, there is interest from members of the working group to have some of these issues revisited in the future by the W3C or other entities as appropriate.

2 Compliance Requirements

This specification is a representation of a rough, inclusive consensus from the Harmonization WG -- meaning that which is specified is recommended as a minimal set of terms. The recommendation and requirements are offset in a colored table. *Requirements are expressed over variables which the WG thinks values must be defined for in order to be a valid P3P proposal*. Products must support the ability to parse and act upon all the variables defined, though we do not specify the way such values need to be acted upon or presented in a graphical user interface; these are left to implementations and user configuration -- which is addressed in the P3P Implementation Guide.

2.1 Nature of Disclosures

To simplify practice declaration, service providers may **aggregate** any of the disclosures (purposes, recipients, and identifiable use) within a statement over data elements. Service providers MUST make such aggregations as an additive operation. For instance, a site that distributes your age to `ours` (ourselves and our agents), but distributes your zip code to `published` (unrelated third parties or public fora), MAY say they distribute your name and zip code to `ours` and `published`. Such a statement appears to distribute more data than actually happens. It is up to the service provider to determine if their disclosure deserves specificity or brevity.

Also, one must always disclose all options that apply. Consider a site with the sole purpose of collecting information for the purposes of `contact` (Contacting Visitors for Marketing of Services or Products). Even though this is considered to be for the `current` (Completion and Support of Current Activity) purpose, the site must state both `contact` and `current` purposes. Consider a site which distributes information to `ours` in order to redistribute it to `published`, the site must state both `ours` and `published` recipients.

3 Definitions

Equable Practice

A practice that is very similar to another in that the purpose, recipients, and identifiable use are the same or more constrained than the original, and the other disclosures are not substantially different. For example, two sites with otherwise similar practices that follow different -- but similar -- sets of industry guidelines.

Identifiable Use

The use of information relating to an individual that identifies that individual -- this may include linking information with personally identifiable information from other sources or combining information so as to infer a person's identity.

Personally Identifiable Data

Any information relating to an identified or identifiable individual. Note that this vocabulary uses a broader term -- Identifiable Use -- that focuses on the way information is used.

Purpose

The reason(s) for data collection and use.

Practice / Statement

The set of disclosures and (optional) solicitations regarding data usage, including purpose, identifiable use, recipients and other disclosures.

Proposal

A collection of one or more privacy statements together with information asserting the identity, URI, assurances, and disclosures of the service covered by the proposal.

Service Provider (Data Controller, Legal Entity)

The person or organization which offers information, products or services from a Web site, collects information, and is responsible for the representations made in a practice statement.

4 Data Categories

A data category is a quality of a data element that may be used by the user's agent to determine what type of element is under discussion.

Status Optional: Service providers MAY use data categories to describe data elements or data sets. If a service provider requires a representation of data that is not otherwise referenceable in an easily understood way, we recommend the following terms be used according to their corresponding definitions.

Physical Contact Information `[physical]`

Information that allows an individual to be contacted or located in the physical world -- such as phone number or address.

Online Contact Information `[online]`

Information that allows an individual to be contacted or located on the Internet -- such as email. Often, this information is independent of the specific computer used to access the network. (See Computer Information)

Unique Identifiers `[uniqueid]`

Non-financial identifiers issued for purposes of consistently identifying the individual -- such as SSN or Web site IDs.

Financial Account Identifiers `[financial]`

Identifiers that tie an individual to a financial instrument, account, or payment system -- such as a credit card or bank account number.

Computer Information `[computer]`

Information about the computer system that the individual is using to access the network -- such as the IP number, domain name, browser type or operating system.

Navigation and Click-stream Data `[navigation]`

Data *passively* generated by *browsing* the Web site -- such as which pages are visited, and how long users stay on each page.

Interactive Data `[interactive]`

Data *actively* generated from or reflecting *explicit interactions* with a service provider through its site -- such as queries to a search engine, logs of account activity, or purchases made on the Web.

Demographic and Socio-economic Data `[demograph]`

Data about an individual's characteristics -- such as gender, age, and income.

Preference Data `[pref]`

Data about an individual's likes and dislikes -- such as favorite color or musical tastes.

Content `[content]`

The words and expressions contained in the body of a communication -- such as the text of email, bulletin board postings, or chat room communications.

State Management Mechanisms `[state]`

Mechanisms for maintaining a stateful session with a user or automatically identifying users who have visited a particular site or accessed particular content previously -- such as HTTP cookies.

Other `[other]`

Other types of data not captured by the above definitions. (A human readable explanation should be provided in these instances.)

* Note: The Computer, Navigation, Interactive and Content categories can be distinguished as follows. The Computer category includes information about the user's computer including IP address and software configuration. Navigation data describes actual user behavior related to browsing. When an IP address is stored in a log file with information related to browsing activity, both the Computer category and the Navigation category should be used. Interactive Data is data actively solicited to provide some useful service at a site beyond browsing. Content is information exchanged on a site for the purposes of communication.

The Other category should be used only when data is requested that does not fit into any other category.

5 Purposes Defined

The following specifies and defines a set of six purposes for data processing relevant to the Web.

> **Status Required**: Service providers MUST use the following terms to explain the purpose of data collection. Service providers MUST disclose *all that apply*. If a service provider does not disclose that a data element will be used for a given purpose, that is a representation that data will not be used for that purpose. Service providers that disclose that they use data for "other" purposes MUST provide human readable explanations of those purposes.

Completion and Support of Current Activity `[current]`

The use of information by the service provider to complete the activity for which it was provided, such as the provision of information, communications, or interactive services -- for example to return the results from a Web search, to forward email, or place an order.

Web Site and System Administration [admin]

The use of information solely for the technical support of the Web site and its computer system. This would include processing computer account information, and information used in the course of securing and maintaining the site.

Customization of Site to Individuals [custom]

The use of information to tailor or modify the content or design of the site to the particular individual.

Research and Development [research]

The use of information to enhance, evaluate, or otherwise review the site, service, product, or market. This does not include personal information used to tailor or modify the content to the specific individual nor information used to evaluate, target, profile or contact the individual.

Contacting Visitors for Marketing of Services or Products [contact]

The use of information to contact the individual for the promotion of a product or service. This includes notifying visitors about updates to the Web site.

Other Uses [other]

The use of information not captured by the above definitions. (A human readable explanation should be provided in these instances.)

6 Purpose Qualifiers

Qualifiers are appended to a purpose to provide additional information on how the purpose is realized with respect to a data element or set of data elements.

Identifiable Use [ID]

Is data used in a way that is personally identifiable -- including linking it with personally identifiable information from other sources? While some data is obviously identifiable (such as full name), other data (such as zip code, salary, or birth date) could allow a person to be identified depending on how it is used. Also, a technically astute person in some circumstances could determine the identity of a user from the IP number in a HTTP log. This requires a specific effort and is based on how that IP number is registered, whether it is used by more than one person on a computer, or if it is dynamically allocated by an internet service provider. Consequently, we refrain from defining any particular data or set of data as personally identifiable and instead focus on whether data is used in an identifiable way. Thus identifiable use applies to data commonly considered to be personally identifiable as well as other data that is used in an identifiable way.

Status Required: Services MUST disclose *one of* the values of the **Identifiable** qualifier.

Non-identifiable [nonid]

Identifiable [id]

Recipients (Domain of Use) [RECPNT]

The **recipients** defines an organizational area, or domain, beyond the service provider and its agents where data may be distributed.

Status Required: Services must disclose *all the* **Recipients** *that apply*.

Comment: Creating a set of values which are simple, informative to the user, and accurate for service provider representations is very challenging and the WG is not completely satisfied with the results. For instance, the issue of transaction facilitators, such as shipping or payment processors, who are necessary for the completion and support of the activity but may follow different practices was problematic. As it stands, such organizations should be represented in whichever category most accurately reflects their practices with respect to the original service provider.

Ourselves and/or our agents [ours]

Ourselves and our agents. We define an agent in this instance as a third party that processes data only on behalf of the service provider for the completion of the stated purposes. (e.g., The service provider and its printing bureau which prints address labels and does nothing further with the information.)

Organizations following our practices [same]

Organizations who use the data on their own behalf under equable practices. (e.g., Consider a service provider that grants the user access to collected personal information, they also provide it to a partner who uses it once but discards it. Since the recipient, who has otherwise similar practices, cannot grant the user access to information that it discarded, they are considered to have equable practices.)

Organizations following different practices [other]

Organizations that are constrained by and accountable to the original service provider, but may use the data in a way not specified in the service provider's practices. (e.g. The service provider collects data that is shared with a partner who may use it for other purposes. However, it is in the service providers interest to ensure that the data is not used in a way that would be considered abusive to the users' and its own interests.)

Unrelated third parties or public fora [published]

Organizations or fora whose data usage practices are not known by the original service provider. (e.g. data is provided as part of a commercial CD-ROM directory, or it is posted on a public on-line Web directory.)

7 General Disclosures

The following are general disclosures about the policies of the service provider. Further information on the policies would be found at the **discURI**.

Access to Identifiable Information [ACCESS] the ability of the individual to view identifiable information and address questions or concerns to the service provider.

Status Required: Service providers must disclose *all* **Access** capabilities *that apply*. The methods of access is not specified. This disclosure applies to the identifiable use disclosure. Any disclosure is not meant to imply that access to all data is possible, but that some of the data may be accessible and that the user should communicate further with the service provider to determine what capabilities they have.

Comment: Service providers may also wish to provide capabilities to access to information collected through means other than the Web at the **discURI.** However, the scope of P3P statements are limited to data collected through HTTP or other Web transport protocols. Also, if access is provided through the Web we recommend the use of strong authentication and security mechanisms for such access, however security issues are outside the scope of this document.

Identifiable Data is Not Used `[nonid]`

[this should be consistent with the use of the identifiable qualifier].

Identifiable Contact Information `[contact]`

access is given to identifiable online and physical contact information (e.g., users can access things such as a postal address).

Other Identifiable Information `[other_ident]`

access is given to other information linked to an identifiable person. (e.g., users can access things such as their online account charges).

Indentifiable Contact Information and Other Identifiable Information `[contact_and_other]`

access is given to identifiable online and physical contact information aw well as to other information linked to an identifiable person.

None `[none]`

no access to identifiable information is given.

Assurance (Accountability)

Does the site have an **assuring party** that attests that the service will abide by its proposal, follows guidelines in the processing of data, or other relevant assertions. Assurance may come from the service provider or an independent assuring party.

Status Required (but specified elsewhere): A required version of this disclosure is implemented through the assurance field, defined in the P3P1.0 specification.

Comment: We expect this field can be used in a number of ways, from representing that one's privacy practices are self assured, audited by a third party, or under the jurisdiction of a regulatory authority.

Other_Disclosures `[OTHER]`

Are Disclosures Made with respect to the following:

> **Status Optional**: If a site wishes to signify in a proposal that it makes a disclosure about change_agreement, or retention, it may do so with the following. No disclosure means that the service provider makes no representation of a policy on that topic.
>
> **Comment**: Some members of the working group felt that 1) disclosures could be made about other topics such as security (see the purpose section), 2) more specific values should be provided, and 3) that such disclosures should be required. However, a strong consensus for this could not be reached in the available time.

Change Agreement `[change_agreement]`

Does the service provider make a disclosure regarding the capability for the user to cancel, or renegotiate the existing agreement at a future time?

Retention `[retention]`

Does the service provider make a disclosure on how long data is retained?

8 References

[P3P]

Marchiori M. and Jaye D. Platform for Privacy Preferences (P3P) Syntax Specification. World Wide Web Consortium. ?-?-1998 (Working Draft)

[**RDF**]

O. Lassila, R. Swick. " Resource Description Framework (RDF) Model and Syntax Specification," World Wide Web Consortium. 22 February 1999 (Recommendation)

[XML-names]

T. Bray, D. Hollander, A. Layman. "Namespaces in XML." World Wide Web Consortium. 14-January-1999. (Recommendation).

[XML]

T. Bray, J. Paoli, C. M. Sperberg-McQueen. "Extensible Markup Language (XML) 1.0 Specification," World Wide Web Consortium. 10-February-1998. (Recommendation)

9 Acknowledgements (Non-normative)

- Liz Blumenfeld, America Online
- Ann Cavoukian, Information and Privacy Commission/Ontario
- Scott Chalfant, Matchlogic

- Lorrie Cranor, AT&T
- Jim Crowe, Direct Marketing Association
- Josef Dietl, World Wide Web Consortium
- David Duncan, Information and Privacy Commission/Ontario
- Melissa Dunn, Microsoft
- Patricica Faley, Direct Marketing Association
- Marit Köhntopp, Privacy Commissioner of Schleswig-Holstein, Germany
- Tony LAM, Hong Kong Privacy Commissioner's Office
- Tara Lemmey, Narrowline
- Jill Lesser, America Online
- Steve Lucas, Matchlogic
- Deirdre Mulligan, Center for Democracy and Technology
- Nick Platten, Data Protection Consultant (formerly of DG XV, European Commission)
- Joseph Reagle, World Wide Web Consortium
- Ari Schwartz, Center for Democracy and Technology
- Jonathan Stark, TRUSTe